Künstliche Intelligenz – Übung 12

Marius Frinken

Friedrich-Alexander-Universität Erlangen–Nürnberg

30.01.2019

Organizational

Homework 11

Homework 12

Misc: Questions, Anecdotes & etc

Organizational

Personal information

my email address: marius.frinken@fau.de

PGP encrypted mails are preferred!

my PGP fingerprint:

`F4BD 7ED4 96A5 9BA6 9FD6 901C 1EEC 9B1B 8CD5 3DA1`

EVALUATION

THANK YOU, 14 PARTICIPANTS!

As promised: Further Info about 10.2

https://fsi.cs.fau.de/forum/thread/16996-Question-regarding-Problem-10-2-and-Challenge

Exam

Date: https://fsi.cs.fau.de/forum/thread/16990-KI-I-exam-date-not-time-fixed-Feb-11

Homework 11

Problem 11.1 & 11.2

(show solution)

As before:

- stick to the notation of the lecture!
- 11.2.1 was a trap ;)

Problems 11.3 - common error(s)

WRONG

$\exists Y. \neg P(Y)$	
$\neg P(A)$	$\exists E$ on ln.9
$P(A)$	assumption

Problems 11.3 - common error(s)

WRONG

$\exists Y . \neg P(Y)$	
$\neg P(A)$	$\exists E$ on ln.9
$P(A)$	assumption

RIGHT

$\exists Y . \neg P(Y)$	
$\neg P(A)$	Assumption ~~$\exists E$ on ln.9~~
$P(A)$	assumption

+ do the ∃E later down the proof

Homework 12

Recap Unification I

Unification (Definitions)

- Problem: For given terms **A** and **B** find a substitution σ, such that $\sigma(\mathbf{A}) = \sigma(\mathbf{B})$.
- **Notation 1.13.** We write term pairs as $\mathbf{A} =^? \mathbf{B}$ e.g. $f(X) =^? f(g(Y))$
- Solutions (e.g. $[g(a)/X], [a/Y]$, $[g(g(a))/X], [g(a)/Y]$, or $[g(Z)/X], [Z/Y]$) are called unifiers, $\mathbf{U}(\mathbf{A} =^? \mathbf{B}) := \{\sigma \mid \sigma(\mathbf{A}) = \sigma(\mathbf{B})\}$
- Idea: find representatives in $\mathbf{U}(\mathbf{A} =^? \mathbf{B})$, that generate the set of solutions
- **Definition 1.14.** Let σ and θ be substitutions and $W \subseteq \mathcal{V}_\iota$, we say that a substitution σ is more general than θ (on W write $\sigma \leq \theta[W]$), iff there is a substitution ρ, such that $\theta = \rho \circ \sigma[W]$, where $\sigma = \rho[W]$, iff $\sigma(X) = \rho(X)$ for all $X \in W$.
- **Definition 1.15.** σ is called a most general unifier of **A** and **B**, iff it is minimal in $\mathbf{U}(\mathbf{A} =^? \mathbf{B})$ wrt. $\leq [\text{free}(\mathbf{A}) \cup \text{free}(\mathbf{B})]$.

Recap Unification II

Unification (Equational Systems)

- Idea: Unification is equation solving.
- **Definition 1.16.** We call a formula $\mathbf{A}^1 =^? \mathbf{B}^1 \wedge \ldots \wedge \mathbf{A}^n =^? \mathbf{B}^n$ an equational system iff $\mathbf{A}^i, \mathbf{B}^i \in wff_\iota(\Sigma_\iota, \mathcal{V}_\iota)$.
- We consider equational systems as sets of equations ($\wedge$ is ACI), and equations as two-element multisets ($=^?$ is C).

Recap Unification III

Unification Algorithm

- **Definition 1.21.** Inference system $\mathcal{U}$

$$\frac{\mathcal{E} \wedge f(\mathbf{A}^1, \ldots, \mathbf{A}^n) =^? f(\mathbf{B}^1, \ldots, \mathbf{B}^n)}{\mathcal{E} \wedge \mathbf{A}^1 =^? \mathbf{B}^1 \wedge \ldots \wedge \mathbf{A}^n =^? \mathbf{B}^n} \mathcal{U}\,\mathrm{dec} \qquad \frac{\mathcal{E} \wedge \mathbf{A} =^? \mathbf{A}}{\mathcal{E}} \mathcal{U}\,\mathrm{triv}$$

$$\frac{\mathcal{E} \wedge X =^? \mathbf{A} \quad X \notin \mathrm{free}(\mathbf{A}) \quad X \in \mathrm{free}(\mathcal{E})}{[\mathbf{A}/X](\mathcal{E}) \wedge X =^? \mathbf{A}} \mathcal{U}\,\mathrm{elim}$$

- ***Lemma 1.22.*** $\mathcal{U}$ is correct: $\mathcal{E} \vdash_{\mathcal{U}} \mathcal{F}$ implies $\mathbf{U}(\mathcal{F}) \subseteq \mathbf{U}(\mathcal{E})$
- ***Lemma 1.23.*** $\mathcal{U}$ is complete: $\mathcal{E} \vdash_{\mathcal{U}} \mathcal{F}$ implies $\mathbf{U}(\mathcal{E}) \subseteq \mathbf{U}(\mathcal{F})$
- ***Lemma 1.24.*** $\mathcal{U}$ is confluent: the order of derivations does not matter
- ***Corollary 1.25.*** First-Order Unification is unitary: i.e. most general unifiers are unique up to renaming of introduced variables.
- Proof Sketch: the inference system $\mathcal{U}$ is trivially branching □

Unification Tips

- **not unifiable**: if $f(...) =_? g(...)$ (also think about constants!)
- **not unifiable**: if $X =_? f(..., X, ...)$
- think about the terms/propositions as trees:
  find a tree **C**, s.t. you can come from trees **A** and **B** to **C** via substitutions (remember to apply your substitution to the **whole trees**, in order to keep every thing consistent!)

Recap First-Order Resolution

First-Order Resolution (CNF)

- **Definition 2.1.** The Conjunctive Normal Form Calculus CNF^1 is given by the inference rules of CNF^0 extended by

$$\frac{(\forall X.\mathbf{A})^{\mathsf{T}} \vee \mathbf{C} \quad Z \notin (\mathrm{free}(\mathbf{A}) \cup \mathrm{free}(\mathbf{C}))}{[Z/X](\mathbf{A})^{\mathsf{T}} \vee \mathbf{C}}$$

$$\frac{(\forall X.\mathbf{A})^{\mathsf{F}} \vee \mathbf{C} \quad \{X_1, \ldots, X_k\} = \mathrm{free}(\forall X.\mathbf{A})}{[f_n^k(X^1, \ldots, X^k)/X](\mathbf{A})^{\mathsf{F}} \vee \mathbf{C}}$$

$CNF^1(\Phi)$ is the set of all clauses that can be derived from Φ.

- **Definition 2.2 (First-Order Resolution Calculus).** First-order resolution is a refutation calculus that manipulates formulae in conjunctive normal form. $\mathcal{R}^1$ has two inference rules

$$\frac{\mathbf{A}^{\mathsf{T}} \vee \mathbf{C} \quad \mathbf{B}^{\mathsf{F}} \vee \mathbf{D} \quad \sigma = \mathbf{mgu}(\mathbf{A}, \mathbf{B})}{\sigma(\mathbf{C}) \vee \sigma(\mathbf{D})} \qquad \frac{\mathbf{A}^{\alpha} \vee \mathbf{B}^{\alpha} \vee \mathbf{C} \quad \sigma = \mathbf{mgu}(\mathbf{A}, \mathbf{B})}{\sigma(\mathbf{A}) \vee \sigma(\mathbf{C})}$$

Howto skip the first annoying part of CNF[1]

1. Put a $\neg$ in front of your formula and label it with T
2. "Drag" the $\neg$ thorough all quantifiers (using the definitions of $\forall, \exists$)
3. Replace variables bound by existential quantifiers with skolem functions depending on all variables that are bound by a forall quantifier on the left side of the respective existential quantifier
4. drop all $\forall$

Example:

(0) $\exists X \forall Y.A(X,Y)$

(1) $\neg\exists X \forall Y.A(X,Y)^T$

(2) $\forall X \exists Y.\neg A(X,Y)^T$

(3) $\forall X.[f(X)/Y](\neg A(X,Y))^T$

(3) $\forall X.\neg A(X,f(X))^T$

(4) $\neg A(X,f(X))^T$

Howto skip the first annoying part of *CNF*[1]

Obviously, the formula has to be in **prenex normal form**! (all quantifiers at the beginning)

Convince yourself, why the last slide works!
(multiple application of the rules of the slides + definitions of $\forall, \exists$)

for details: see https://fsi.cs.fau.de/dw/_media/pruefungen/hauptstudium/ki1-summary-ws1819.pdf page 6

Recap First-Order Tableaux I

Free variable Tableaux ($\mathcal{T}_1^f$)

- Refutation calculus based on trees of labeled formulae
  - $\mathcal{T}_0$ (propositional tableau rules) plus
  - Quantifier rules:

$$\frac{\forall X.\mathbf{A}^{\mathsf{T}} \quad Y \mathit{new}}{[Y/X](\mathbf{A})^{\mathsf{T}}}\mathcal{T}_1^f{:}\forall \qquad \frac{\forall X.\mathbf{A}^{\mathsf{F}} \quad \mathrm{free}(\forall X.\mathbf{A}) = \{X^1, \ldots, X^k\} \quad f \in \Sigma_k^{sk}}{[f(X^1, \ldots, X^k)/X](\mathbf{A})^{\mathsf{F}}}\mathcal{T}_1^f{:}\exists$$

  - Generalized cut rule: $\mathcal{T}_1^f{:}\bot$ instantiates the whole tableau by σ.

$$\frac{\begin{array}{l}\mathbf{A}^{\alpha}\\ \mathbf{B}^{\beta}\end{array} \quad \alpha \neq \beta \quad \sigma(\mathbf{A}) = \sigma(\mathbf{B})}{\bot : \sigma}\mathcal{T}_1^f{:}\bot$$

  Advantage: no guessing necessary in $\mathcal{T}_1^f{:}\forall$-rule

- New: find suitable substitution (most general unifier)

Recap First-Order Tableaux II

- no rule for $\exists$, so you have to rewrite first!
- When using the cut rule, you have to apply the substitution on the whole proof!

STRIPS

STRIPS planning task This is a encoding of a planning problem using a 4-tuple $\Pi = \langle P, A, I, G \rangle$ where

- P is a finite set of facts
- A is a finite set of actions, each given as a triple $\langle pre_a, add_a, del_a \rangle$. The components of the triple are called preconditions, add-list and delete-list
- $I \subseteq P$ is the initial state
- $G \subseteq P$ is the goal.

Example I

P :
$RobotX(n), 0 \leq n \leq 5$
$RobotY(n), 0 \leq n \leq 5$
$Marked_{x,y}(b), b \in \{true, false\}, 0 \leq x, y \leq 5$
I :
$\{RobotX(0), RobotY(0), Marked_{x,y}(false)$ for all $0 \leq x, y \leq 5\}$
G :
$\{Marked_{5,5}(true)\}$

A : (only small example)
Forall $0 \leq X, X_{new} \leq 5$ **and** $|X - X_{new}| = 1$: $Walk_{X,X_{new}}$:
pre: $\{RobotX(X)\}$ add: $\{RobotX(X_{new})\}$ del: $\{RobotX(X)\}$

Forall $0 \leq X, Y \leq 5$: $Mark_{X,Y}$:
pre: $\{Marked_{X,Y}(false), RobotX(X), RobotY(Y)\}$
add: $\{Marked_{X,Y}(true)\}$ del: $\{Marked_{X,Y}(false)\}$

Example II – a little bit different

P :
$RobotX(n), 0 \leq n \leq 5$
$RobotY(n), 0 \leq n \leq 5$
$Marked_{x,y}, 0 \leq x, y \leq 5$
I :
$\{RobotX(0), RobotY(0)\}$
G :
$\{Marked_{5,5}\}$

A : (only small example)
Forall $0 \leq X, X_{new} \leq 5$ **and** $|X - X_{new}| = 1$: $Walk_{X,X_{new}}$:
pre: $\{RobotX(X)\}$ add: $\{RobotX(X_{new})\}$ del: $\{RobotX(X)\}$

Forall $0 \leq X, Y \leq 5$: $Mark_{X,Y}$:
pre: $\{RobotX(X), RobotY(Y)\}$
add: $\{Marked_{X,Y}\}$ del: $\{\}$

Differences?

- Example II shows that *Marked* can realized without a boolean value, the sheer presence of a fact is enough here
- Example II uses the properties of mathematical sets
- Example II allows for applying *Mark*, even if the field is already marked (adding an already existing element to a **set** does nothing)
- Both are valid i.e. in the exam/assignments, you yourself can make your life easier/harder

Misc: Questions, Anecdotes & etc

Questions?