Last Name: First Name:

Matriculation Number:

Exam
Logic-Based Natural Language Semantics

April 10, 2025

Please ignore the QR codes; do not write on them, they are for grading support

	To be used for grading, do not write here												
prob.	1.1	1.2	1.3	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	Sum	grade
total	3	3	4	3	4	8	7	12	9	4	7	64	
reached													

In the course Artificial Intelligence I/II we award bonus points for the first student who reports a factual error in an old exam. (Please report spelling/formatting errors as well.)

1 Foundations

Problem 1.1 (PL^{pq} Models) 3 Points

Objective: understand first-order model

Objective: apply interpretation

Consider the first-order signature

$$\Sigma_0^f = \{x\}$$
$$\Sigma_1^f = \{f\}$$
$$\Sigma^p{}_1 = \{p\}$$

the PL^{pq} formula

$$\varphi := p(x) \wedge \neg p(f(x)) \wedge p(f(f(x)))$$

and the first-order model $\langle \mathcal{D}, \mathcal{I} \rangle$ where

$$\mathcal{D} = \{a, b\}$$
$$\mathcal{I}(x) = a$$
$$\mathcal{I}(p) = \{a\}$$

For which value of $\mathcal{I}(f)$ is φ true in $\langle \mathcal{D}, \mathcal{I} \rangle$?

Solution:

$$\mathcal{I}(f) = \{a \mapsto b, b \mapsto a\}$$

Problem 1.2 (Beta reduction) 3 Points

Objective: apply beta reduction

Apply β-reduction to the following term until it is fully reduced to normal form:

$$(\lambda x.x\ (\lambda y.z))\ (\lambda p.p\ a\ b)$$

Solution:

$$(\lambda x.x\ (\lambda y.z))\ (\lambda p.p\ a\ b)$$
$$\rightsquigarrow_\beta (\lambda p.p\ a\ b)\ (\lambda y.z)$$
$$\rightsquigarrow_\beta (\lambda y.z)\ a\ b$$
$$\rightsquigarrow_\beta z\ b$$

Problem 1.3 (Simply Typed) 4 Points

Objective: apply well typed formula

Consider the following untyped λ term:

$$T := \lambda x.a\ (x\ a)$$

We want to add types for x and a to get a simply typed λ term. Which of the following statements are meaningful and true?

- ☐ T is well-typed, no matter what types we assign to x and a
- ☐ T is well-typed if x has type $\gamma \to \beta$ and a has type γ

- ☐ T is well-typed if x has type $(\gamma \to \gamma) \to \gamma$ and a has type γ
- ☑ T is well-typed if x has type $(\gamma \to \gamma) \to \gamma$ and a has type $\gamma \to \gamma$
- ☑ If we know the type of a, we can uniquely determine the type of x
- ☑ If we know the type of x, we can uniquely determine the type of a
- ☐ T cannot be typed because the definition of x is not a function
- ☐ T cannot be typed because the definition is recursive

2 The Method of Fragments

Problem 2.1 (Natural Language Semantics) 3 Points

In the method of ***A*** *, we specify the syntax and semantics of increasingly complex* ***B*** *of natural language. The syntax is specified in a* ***C*** .

Objective: understand fragment

Blank **A**:

- ☑ fragments
- ○ entailment relation
- ○ truth conditions

Blank **B**:

- ☑ subsets
- ○ intersections
- ○ supersets

Blank **C**:

- ○ tableau machine
- ☑ grammar
- ○ logic

Problem 2.2 (Overgenerating grammar)

Consider the following grammar with start symbol A:

$A1: A \to A$, and, A,
$A2: A \to D, E$,
$B1: B \to$ John,
$B2: B \to$ Mary,
$C1: C \to$ dog,
$C2: C \to$ cat,
$D1: D \to B$,
$D2: D \to C$,
$D3: D \to$ the, D,
$E1: E \to$ sleeps,

1. Provide an example sentence that illustrates the over-generation of this grammar. 2 Points

Objective: understand over-generate

Solution: Many options, e.g.:

1. *the the cat sleeps*
2. *cat sleeps*
3. *the John sleeps*

---

2. How can the grammar be fixed to avoid over-generation? State which rules should be removed and what should be added. Your grammar should still cover all correct sentences from the original grammar.

2 Points
Objective: apply grammar

Solution: Remove $D3$ and replace $D2$ with $D2: D \to \text{the}, C$,

---

Problem 2.3 (Or-Fragment)
Consider the following grammar with start symbol S on the left and and the semantics construction rules on the right:

Objective: create grammar
Objective: create semantics construction

$S1: S \to \text{NP}, VP$,	$T1: [X_{\text{NP}}, Y_{VP}]_S \rightsquigarrow Y'(X')$
$N1: \text{NP} \to \text{John}$,	$T2: [\text{John}]_{\text{NP}} \rightsquigarrow j$
$N2: \text{NP} \to \text{Mary}$,	$T3: [\text{Mary}]_{\text{NP}} \rightsquigarrow m$
$V1: VP \to \text{sleeps}$,	$T4: [\text{sleeps}]_{VP} \rightsquigarrow s$
$V2: VP \to \text{runs}$,	$T5: [\text{runs}]_{VP} \rightsquigarrow r$

1. Extend the grammar and the semantics construction rules so that sentences like *John runs or Mary sleeps* are supported.

3 Points

Solution:

$$S2: S \to S, \text{or}, S$$

$$T6: [X_S, \text{or}, Y_S]_S \rightsquigarrow (X' \vee Y')$$

---

2. Extend the grammar and the semantics construction rules so that sentences like *John runs or sleeps* are supported.

3 Points

Solution:

$$V3: VP \to VP, \text{or}, VP$$

$$T7: [X_{VP}, \text{or}, Y_{VP}]_{VP} \rightsquigarrow (\lambda t.X'(t) \vee Y'(t))$$

---

3. To support sentences like *John or Mary runs*, we need type raising. If we have a type raised semantics construction, what should be the meaning of *John or Mary*? (i.e. what should be the result of the semantics construction?)

2 Points
Objective: apply type raising

Solution:

$$\lambda p.p(j) \vee p(m)$$

Problem 2.4 (The meaning of "Only")
Consider the following two sentences:

1. *Only Peter runs.*
2. *Only dogs bark.*

1. For both sentences, write down the interpretation as a first-order formula

4 Points

Objective: apply PREDLOG

Solution:

1. $\text{runs}(\text{peter}) \wedge (\forall x.\neg x = \text{peter} \Rightarrow \neg\text{runs}(x))$
2. $(\forall x.\text{dog}(x) \Rightarrow \text{bark}(x)) \wedge (\forall y.\neg\text{dog}(y) \Rightarrow \neg\text{bark}(y))$

2. The word *only* is handled differently in the two cases above, for both cases give the interpretation as a $\text{HOL}^{\rightarrow}$ formula. Give the types of the bound variables:

3 Points

Objective: apply beta expansion

Solution:

1. $\lambda P.\lambda Q.Q(x) \wedge (\forall x.\neg x = P \Rightarrow \neg Q(x))$
   Types: $P, x : \iota$ and $Q : \iota \to o$
2. $\lambda P.\lambda Q.(\forall x.P(x) \Rightarrow Q(x)) \wedge (\forall y.\neg P(y) \Rightarrow \neg Q(y))$.
   Types: $P, Q : \iota \to o$ and $x, y : \iota$.

Problem 2.5 (Tableau Machine)
Consider the sentence
The dog chased the cat. It climbed up the tree.

Objective: apply tableau machine
Objective: apply description operator
Objective: apply reading

1. Construct a model generation tableau to represent the following discourse, incorporating only information contained in the sentences. Make sure that you use a suitable (compositional) representation of definite descriptions.

4 Points

Solution:

$$\boxed{\mathit{chase}(\iota\ \text{dog}, \iota\ \text{cat})}$$
$$\boxed{\mathit{climb}(X, \iota\ \text{tree})}$$
$$\mathit{climb}(\iota\ \text{dog}, \iota\ \text{tree})^{\mathsf{T}} \mid \mathit{climb}(\iota\ \text{cat}, \iota\ \text{tree})^{\mathsf{T}} \mid \mathit{climb}(\iota\ \text{tree}, \iota\ \text{tree})^{\mathsf{T}}$$

2. How many possible readings are predicted?

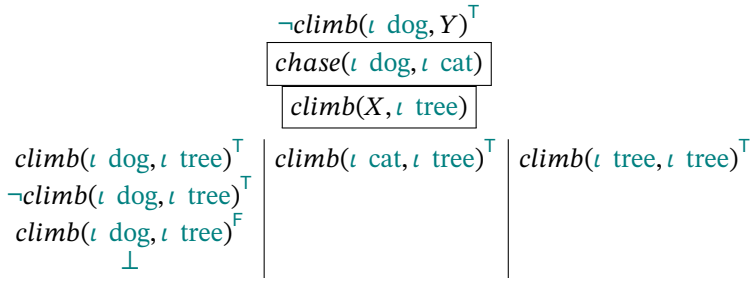
1 Points

Solution: Three (no branch closes)

3. Now modify the tableau by including a representation of the world knowledge that the dog does not climb up anything.

3 Points

Solution:



4. How would the tableau machine for natural language understanding deal with the situation where we have multiple open branches? 2 Points

Solution: It would choose a preferred branch and focus on the reading/model it represents by adding new information to that branch and saturating the subtableau – backtracking on the branch choice if the new subtableau closes.

5. Finally, what do we have to add as world knowledge (based on the concept that trees are plants; do not just state that "trees – like dogs above – do not climb anything) to make sure that we only get one reading as intuitively expected. 2 Points
You do not have to draw the tableau, just state the world knowledge.

Solution:

1. Trees are plants.
2. Plants cannot move.
3. If something cannot move, it cannot climb.

3 Topics in Semantics

Problem 3.1 (DRT Representation and Semantics)
Consider the following discourses:

Objective: apply anaphoric potential
Objective: apply direct semantics
Objective: apply DRT translation

1. *If a farmer[1] owns a donkey[2], he$_1$ beats it$_2$ with a stick.*
2. *A man hit Mary, or a man bit Mary. She is crying.*

1. Represent them in DRT[1]. 3 Points

Solution:

1. $D_1 :=$

X, Y	⇒	Z
farmer(X)		stick(Z)
donkey(Y)		beatwith(X, Y, Z)
own(X, Y)		

[1] You can use top-level discourse referents for Mary, or more simply just an individual constant. In the second case, you have to be a bit imaginative to resolve the anaphor.

2. The following two representations are OK:
   - $D_2 := (\boxed{\begin{array}{|l} X \\ \hline \text{beat}(X,M) \vee \text{bites}(X,M) \end{array}} \otimes \boxed{\begin{array}{|l} Y \\ \hline \text{cries}(Y) \\ Y = M \end{array}})$,
   - or the already-merged version $D_3 := \boxed{\begin{array}{|l} X, Y \\ \hline \text{beat}(X,M) \vee \text{bites}(X,M) \\ \text{cries}(Y) \\ Y = M \end{array}}$.

---

2. Compute the translations to first-order logic 3 Points

Solution:

1. $\forall X, Y.\text{farmer}(X) \wedge \text{donkey}(Y) \wedge \text{own}(X,Y) \Rightarrow (\exists Z.\text{beatwith}(X,Y,Z))$
2. $\exists X, Y.(\text{beat}(X,M) \vee \text{bites}(X,M)) \wedge \text{cries}(Y) \wedge Y = M.$

---

3. Compute the direct semantics for them. What is the anaphoric potential? 3 Points

---

Solution:

1. $\mathcal{I}^{\delta}_{\varphi}(D_1) = (\emptyset, \{\psi \mid \begin{array}{l} \psi[X,Y]\varphi, \psi(X) \in \mathcal{I}(\text{farmer}), \psi(Y) \in \mathcal{I}(\text{donkey}), \\ (\psi(X), \psi(Y)) \in \mathcal{I}(\text{own}), \text{and there is a } \rho[Z]\psi \text{ with} \\ \rho(Z) \in \mathcal{I}(\text{stick}) \text{ and } \langle \rho(X), \rho(Y), \rho(Z) \rangle \in \mathcal{I}(\text{beatwith}) \end{array} \})$
2. We have only defined direct semantics for $\otimes$-reduced DRSes, so we only treat D_3:

$$\mathcal{I}^{\delta}_{\varphi}(D_3) = (\{X,Y\}, \{\psi \mid \begin{array}{l} \psi[X,Y]\varphi \text{ and} \\ ((\varphi(X), \mathcal{I}(M)) \in \mathcal{I}(\text{beat}) \text{ or } (\psi(X), \mathcal{I}(M)) \in \mathcal{I}(\text{bites})) \\ \text{and } \psi(Y) \in \mathcal{I}(\text{cries}) \text{ and } \varphi(Y) = \mathcal{I}(M) \end{array} \})$$

Problem 3.2 (Compositionality, Congruence, and Propositional Attitudes) 4 Points

Define the compositionality and congruence principles in general.
Discuss whether they hold when modeling propositional attitudes in natural language via modal logics; give counter-examples if one does not hold.

Objective: understand compositional

Objective: apply congruence principle

Solution: The compositionality principle states that the meaning of a complex expression is a function of the meanings of its sub-expressions, i.e. it only depends on those.
The congruence principle states that whenever A is part of B and A' means just the same as A, replacing A by A' in B will lead to a result that means just the same as B.
Modal logics are prominent meaning theories for propositional attitudes in natural language, most are compositional, but not do not obey congruence. The most prominent counter-example is that the ancient Greeks did not believe (a propositional attitude) that the morning star is the evening star even though the meaning of both is the planet Venus (as we now know).

Problem 3.3 (Repeat in DPL1)
Consider the following program α written in pseudocode

```
var X = 0; var Y = 10
repeat X:=X+1; Y:=Y-1 until X=Y
```

1. Write α in the internal programming language of DL^1.

3 Points
Objective: create first-order program logic

Solution: $X:=0\,;Y:=10\,;*(X:=X-1\,;Y:=Y+1\,;\neg(X=Y)?)\,;(X=Y)?$

---

2. State the partial correctness of α with respect to the specification that X and Y are equal after running α in DL^1.

1 Points
Objective: apply MMnecessary

Solution: $[\alpha]X=Y$.

---

3. State the termination of α in DL^1.

1 Points
Objective: apply MMpossible

Solution: $\langle\alpha\rangle T$.

---

4. Is DL^1 sufficient to fully represent the intended semantics of the program α and thus prove or refute partial correctness? If not, what do we additionally need?

2 Points
Objective: analyze first-order program logic

Solution: No, it is not. We can fully represent the program semantics, but in first-order logic, we cannot fix the semantics of equality, addition, and subtraction. We could pass to "DL^1 with interpreted function and predicate symbols" mentioned in the course, or add an axiomatization $\mathcal{A}$ of these as a precondition to the statement: $\mathcal{A}\Rightarrow[\alpha]X=Y$.
