Last Name: First Name:

Matriculation Number:

Seat:

Exam
—
Logic-Based Natural Language Semantics (LBS)

February, 21. 2023

	To be used for grading, do not write here									
prob.	1.1	1.2	2.1	2.2	2.3	3.1	3.2	3.3	Sum	grade
total	3	3	4	4	6	3	4	6	33	
reached										

The “solutions” to the exam/assignment problems in this document are supplied to give students a starting point for answering questions. While we are striving for helpful “solutions”, they can be incomplete and can even contain errors even after our best efforts.
In any case, grading student’s answers is not a process of simply “comparing with the reference solution”, therefore errors in the “solutions” are not a problem in this case.
If you find “solutions” you do not understand or you find incorrect, discuss this on the course forum and/or with your TA and/notify the instructors. We will – if needed – correct them ASAP.

1 Epistemology

Problem 1.1 (True or False in Epistemology?) 3 pt

Check if the following statements are true or false :

- ☐ Every *observation* is *reproducible*. – Wrong
- ☐ *Knowledge* is always a *belief*. – Correct
- ☐ A *hypothesis* needs to be *minimal*. – Wrong
  No, theories need to be minimal, hypotheses don't

Problem 1.2 (Epistemological Terms and their Relations) 3 pt

Relate the terms

- *phenomenon*,
- *proposition*,
- *hypothesis*

to each other.

2 The Method of Fragments

Problem 2.1 4 pt

In the pipeline of syntactic processing, semantics construction, semantic/pragmatic analysis discussed in the LBS lecture, highlight and explain the role of context-free and compositional methods.

Problem 2.2 4 pt

Why are we often more interested in models rather than proofs in NLP scenarios?

Solution: Communicating a natural language utterance can be interpreted as a way of changing the world model of the hearer to a state, where it is consistent with the utterance and the previous model the hearer entertains.

Problem 2.3 6 pt

When interpreting natural language *utterances*, the three problems *abstraction*, *ambiguity* and *composition* arise. Give an example each. Explain the concept briefly.

Solution:

1. **Abstraction** describes the situation, where more than one utterance can have the same meaning: Synonyms like *car* and *automobile* have the same meaning

2. **Ambiguity** describes the situation, where one utterance can have more than one meaning, e.g. the word *bank* can denote a landscape feature or a financial institution.

3. **Composition** problems arise where the syntactical structure of an utterance does not directly correspond to the structure of the meaning representation. E.g. the sentence *Every student sleeps* has a different syntactic structure than the *first-order logic* representation $\forall x.\mathrm{stud}(x) \Rightarrow \mathrm{sleep}(x)$.

---

3 GLIF

Problem 3.1 (GF-based Translation) 3 pt

How can we use GF to directly translate between natural languages?

Solution: We define concrete grammars $G_1, \ldots, G_n$ (one for each language L_i) that share one abstract grammar G. For a translation between L_i and L_j, we parse using G_i and linearize using G_j.

Problem 3.2 4 pt

You want to add the new operator "there are exactly three" $\exists^3$ to your (first-order) logic. How can you do that using higher order abstract syntax (HOAS) in MMT? Explain the concept of HOAS in general, state the declaration in MMT surface syntax, and explain the intended semantics of the operator.

Solution: HOAS is an approach to represent the syntax of binders. Concretely, we use a (lambda) function (provided by the meta language) to, in this case, represent the $\exists^3$ binder:

```
exists_three : (ι → o) → o | # ∃³1 ❚
```

We can interprete the argument of $\exists^3$, which is a unary predicate, as a set. Then $\exists^3\ S$ is supposed to state that the cardinality of S is 3.

Problem 3.3 (PLNQ Proof Rules) 6 pt

Express the following rules of natural deduction in MMT.

$$\frac{\mathbf{A}\quad \mathbf{B}}{\mathbf{A} \wedge \mathbf{B}}\wedge I \qquad \frac{\begin{array}{c}[\mathbf{A}]^1\\ \hline\hline \mathbf{B}\end{array}}{\mathbf{A} \Rightarrow \mathbf{B}}\Rightarrow I^1 \qquad \frac{\forall X.\mathbf{A}}{[\mathbf{B}/X](\mathbf{A})}\forall E \qquad \frac{\exists X.\mathbf{A} \quad \begin{array}{c}[[c/X](\mathbf{A})]^1\\ \vdots\\ \mathbf{C}\end{array} \quad c \text{ new}}{\mathbf{C}}\exists E^1$$

Solution:

```
andI : {A:o, B:o} ⊢A →⊢B →⊢A∧ B ❚
implI : {A:o, B:o} (⊢A →⊢B) →⊢A⇒ B ❚
forallE : {A:ι→o, B:ι} ⊢∀A →⊢A B ❚
existsE : {A:ι→o, C:o} ⊢∃A →({c:ι} ⊢A c →⊢C) →⊢C ❚
```

Problem 3.4 (Extend a GLIF fragment) 18 pt

Use and extend the GLIF implementation of a small fragment of the English language.

1. How would the sentence "John is happy" be processed by the implementation? Specify the abstract syntax tree and the results of the semantics construction (before and after β-reduction).
2. Extend the grammar (both abstract and concrete) to support more sentences. Concretely, you should add three rules:
   (a) `make_not_S` to support negated sentences like "John isn't happy",
   (b) `cond_S` to support conditional sentences like "John is happy if Mary is lucky",
   (c) `and_Adjective` that combines two adjectives into a new one like "happy and lucky" (for simplicity, we do not use a separate category for adjectival phrases).
3. Complete the semantics construction for the new fragment and add the required logical connectives to the `logic` theory. For example, we expect the following results (your implementation may yield logically equivalent propositions):
   (a) "John isn't happy" $\mapsto \neg h(j)$
   (b) "John isn't happy if Mary is lucky" $\mapsto l(m) \Rightarrow \neg h(j)$
   (c) "Mary isn't happy and lucky" $\mapsto \neg(h(m) \wedge l(m))$

```
abstract Grammar = {
    cat
        S; -- sentence
        Name;
        Adjective;
    fun
        john: Name;
        mary: Name;
        happy: Adjective;
        lucky: Adjective;

        make_S : Name -> Adjective -> S;

        -- Add your code here:
```

```
}
```

```
concrete GrammarEng of Grammar = {
    lincat
        S = Str;
        Name = Str;
        Adjective = Str;

    lin
        john = "John";
        mary = "Mary";
        happy = "happy";
        lucky = "lucky";

        make_S n a = n ++ "is" ++ a;

        -- Add your code here:











}
```

```
theory logic : ur:?LF =
    prop : type | # o ❙
    individual : type | # ι ❙

    and : o → o → o | # 1 ∧ 2 ❙
    impl : o → o → o | # 1 ⇒ 2 ❙

    // Add your code here: ❙




❚
```

```
theory people : ur:?LF =
    include ?logic ❙
    j : ι ❙
    m : ι ❙
    h : ι → o ❙
    l : ι → o ❙
❚
```

```
view SemanticsConstruction : .../Grammar.gf?Grammar -> ?people =
    S = o ❙
```

```
Name = ι
Adjective = ι → o

john = j
mary = m
happy = h
lucky = l

make_S = [n, a] a n

// Add your code here:
```

4 Discourse Semantics

Problem 4.1 (Ambiguity in Event Semantics) 4 pt

Write down all *readings* of the sentence *Peter chases the gangster in the red sportscar.* in *first-order logic* using the *event semantics* approach.

Hint: You can invent any constants and predicates you want.

Solution: There are three readings

1. $\exists e.\exists c.\mathrm{chase}(e) \wedge \mathrm{redsportscar}(c) \wedge \mathrm{ag}(e, \mathrm{peter}) \wedge \mathrm{pat}(e, \mathrm{gangster}) \wedge \mathrm{in}(\mathrm{peter}, c)$
2. $\exists e.\exists c.\mathrm{chase}(e) \wedge \mathrm{redsportscar}(c) \wedge \mathrm{ag}(e, \mathrm{peter}) \wedge \mathrm{pat}(e, \mathrm{gangster}) \wedge \mathrm{in}(\mathrm{gangster}, c)$
3. $\exists e.\exists c.\mathrm{chase}(e) \wedge \mathrm{redsportscar}(c) \wedge \mathrm{ag}(e, \mathrm{peter}) \wedge \mathrm{pat}(e, \mathrm{gangster}) \wedge \mathrm{in}(e, c)$

Problem 4.2 (Dynamic Effects) 4 pt

1. What is the (linguistic) difference between the following two discourses:

   (a) *There is a book that Peter does not own. It is a novel.*

   (b) * *Peter does not own every book. It is a novel.*

   In particular, why is the second one not felicitous (i.e. OK)?

2. What is the problem when we try to model their meaning in *first-order logic*?

Solution:

1. The second discourse is infelicitous as its first sentence does not introduce an antecedent for the pronoun *It* to pick up on.
2. The problem is that that the respective first sentences are logically equivalent: Their translations into *first-order logic* are
   (a) $\exists x.\mathrm{book}(x) \wedge \neg\mathrm{own}(p, x)$
   (b) $\forall x.\mathrm{book}(x) \Rightarrow \mathrm{own}(p, x) \Rightarrow$

   Thus in *first-order logic* we cannot distinguish them.

Problem 4.3 (Modeling a Discourse as a DRS) 6 pt

Given the discourse
A student takes an exam. She is worried about it.

1. Represent the two sentences as separate DRSes.
2. How do you represent anaphora resolution here?
3. What happens if you merge them into into a single DRS.

Hint: You can invent any predicates you want.

Solution:

1. *A student takes an exam.*

U, V
student(U) exam(V) take(U, V)

She is worried about it.

X, Y
worry(X, Y)

2. In anaphor resolution we add two new *conditions* in the second DRS, yielding

X, Y
worry(X, Y) $U = X$ $V = Y$

3. The merge operation makes a *DRS* whose *discourse referents* and *conditions* are the *unions* of the argument *DRSes*.

---

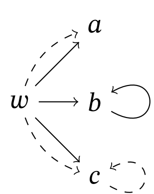
5 Modal Logic

Problem 5.1 5 pt

Given a multimodal logic with two modalities [1] and [2]. Evaluate the following formulae

1. $[1]X$,
2. $\langle 2\rangle X$,
3. $[1]\langle 2\rangle X$,
4. $\langle 1\rangle [2]X$,
5. $\langle 1\rangle(\neg X \wedge \neg\langle 1\rangle X)$.

in world w in the following *Kripke structure* and briefly justify your answer. The solid arrows represent the *accessibility relation* for [1] and the dashed ones for [2]. Use the variable assignment φ with



$$\begin{aligned} \varphi(w,X) &= \mathsf{T} \\ \varphi(a,X) &= \mathsf{T} \\ \varphi(b,X) &= \mathsf{F} \\ \varphi(c,X) &= \mathsf{F} \end{aligned}$$

Solution: The correct answers (without justifications) are:

1. $\mathcal{I}_\varphi^w([1]X) = \mathsf{F}$,
2. $\mathcal{I}_\varphi^w(\langle 2\rangle X) = \mathsf{T}$,
3. $\mathcal{I}_\varphi^w([1]\langle 2\rangle X) = \mathsf{F}$,
4. $\mathcal{I}_\varphi^w(\langle 1\rangle [2]X) = \mathsf{T}$,
5. $\mathcal{I}_\varphi^w(\langle 1\rangle(\neg X \wedge \neg\langle 1\rangle X)) = \mathsf{T}$.
