Automated Software Engineering Exam 1 (December 19, 2024) Group A

Task 1: Short questions 15 points

3.1 We are developing an emergency braking system. At the current state of development, we have a test suite that does not cover some corner cases in the code base. It is challenging for the test engineers to create test inputs for those corner cases (that might be impossible to reach), so we decided to buy a test generator tool. Should we buy a test generator based on random search or symbolic execution? Please explain!

3.2 We want to create a coding assistant tool based on the large language models. Our current goal is to propose the most probable method to call. The current version shows promising results (e.g., for a string manipulation task, it finds the "`substring`" method), but it also lists non-existing interaction methods (e.g., a hallucinated "`substringStartWithCharacter`" method). How to fix this problem?

3.3 We are measuring the performance of a process. We measure that the following throughputs at different arri-val rates (all measured in 1/s). Please explain all your answers!

a) At which arrival rate the system is in balance?
b) What is the maximum throughput?
c) What is the expected utilization at 800/s arrival rate?

Arrival rate	1000	1200	1400	1600	1800
Throughput	1000	1200	1400	1580	437

Task 2: Graph modeling 25 points

a) Write a **Refinery metamodel** based on the following specification:

We would like to use graph modeling and logic to manage and integrate information in a deep-sea hydrothermal vent ecosystem database. The database stores information about the structure of vent fields and the various research expeditions that have gathered data about them. Vent fields are at the meeting of 2 tectonic plates. A vent field can contain between 1 and 50 individual vents, which can either be white smokers or black smokers. Each vent is located on a specific tectonic plate and emits at least 1 mineral. An expedition studies a vent field with at least 1 but at most 10 vessels. A vessel can be either a surface ship or an underwater vehicle. An underwater vehicle may be optionally carried by a surface ship.

Only provide Refinery code and do NOT write Java code or draw an UML class diagram.

Use the following concept names: `BlackSmoker, carriedBy, emits, Expedition, locatedOn, Mineral, plates, studies, SurfaceShip, TectonicPlate, UnderwaterVehicle, Vent, VentField, vents, Vessel, vessels, WhiteSmoker`

b) Draw a **graph model** based on the following data:

The Juan de Fuca vent field is located at the meeting of the Pacific and the Nazca tectonic plates. At the vent field, a black smoker located on the Pacific plate emits sulfur and iron, while a white smoker located on the Nazca plate emits copper. The "Juan de Fuca 2023" expedition visited the vent field with the surface ship Atlantis and the two underwater vehicles Alvin and Jason. Alvin was carried by Atlantis, but Jason approached the vent field on her own.

Only provide a graph model and do NOT write Java or Refinery code.

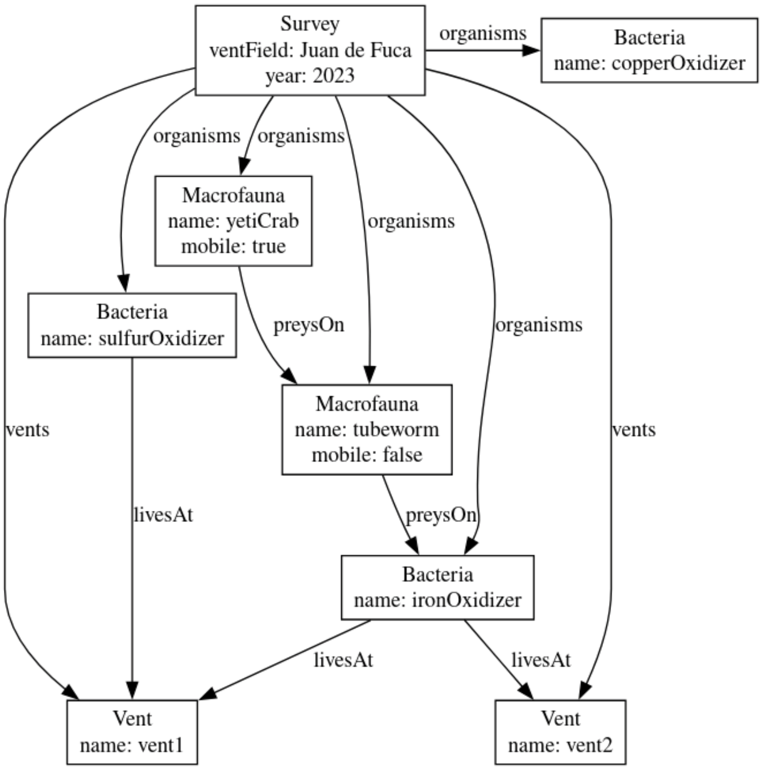
c) According to a domain expert, the following **constraint** holds: *a vent field can only contain vents that are located on some plate that meets at the vent field.* However, the expert suspects that this cannot be enforced by the current version of the metamodel. To confirm, draw a **graph model** that conforms to the metamodel but violates the constraint.

d) Recommend a way to incorporate the constraint mentioned in c) into the metamodel.

Task 3: Textual modeling **30 points**

We would like to create a **textual domain-specific modeling language** to describe the ecosystems found at the surveyed hydrothermal vent fields. The example below shows the desired **concrete syntax** (textual description) **and abstract syntax** (instance graph model) for the language:

```
survey of "Juan de Fuca" 2023:
vents:
  - vent1
  - vent2
bacteria:
  - ironOxidizer around vent1, vent2
  - sulfurOxidizer around vent2
  - copperOxidizer
macrofauna:
  - sessile tubeworm preys on ironOxidizer
  - mobile yetiCrab preys on tubeworm
```



a) Create a **Langium grammar** to parse this language! These declarations are available to you:

```
grammar Ecosystem

hidden terminal WS: /\s+/;
terminal ID: /[_a-zA-Z][\w_]*/;
terminal INT: /\d+/;
terminal STRING: /"[^"]*"/;
```

Provide the rest of the grammar.

b) Create a **Jinja2 template** to generate an observation log in Markdown format. The input of the template is the `Survey` object parsed by the grammar you created in part a). An example observation log is shown below:

```
# Observation log for Juan de Fuca expedition in 2023
## Bacteria observed at vent1
* [ ] ironOxidizer
* [ ] sulfurOxidizer
## Bacteria observed at vent2
* [ ] ironOxidizer
## Macrofauna observed
* [ ] tubeworm sitting in one place
* [ ] yetiCrab moving around
```

To help you in reading the example, we emphasized text coming directly from instance model in **bold**. Make sure to output a heading (`##`) for bacteria observed at each vent and a separate heading for macrofauna. Output the description “`sitting in one place`” for sessile macrofauna and “`moving around`” for mobile macrofauna.

In the instance model, the `type` attribute contains the type of each object (e.g., use `x.type == "Bacteria"` to check if `x` is of type `Bacteria`). Cross-references are encoded as strings (or arrays of strings) equal to the `name` of the referenced object(s).