Programming language concepts

1. **What is ReDoS and what part do ‘Evil Regex’ play?**

   Regular expression Denial of Service (ReDoS) is an attack which makes the server/application to work very slowly and hang for a long time.

   Faulty regular expressions cause ReDoS.

2. **What are the common problems associated with the use of regex? How can these be mitigated?**

   Common problems are:
   - Very few regular expressions fail to compile
   - Some symbols have different meanings in different situations

   To mitigate the problems, faulty regular expressions should be avoided because they reject the acceptable strings and accept unacceptable strings. Regular expression checker tools can also be beneficial.

3. **How and why could regex be used as part of a security solution?**

   Avoid catastrophic backtracking by writing regular expressions which fail fast (i.e.: don’t spend too much time) and are the only unique way to match the string.

What is an ontology?

Ontology refers to set of concepts, data, and entities of a certain domain (subject/discipline). It also includes categories, properties, and relationships related to those entities.

Ontology has been introduced to different domains. So, many domain experts can share information in their fields. Meanwhile, other domain experts can use existing ontologies or even integrate their own into a large one.

The reason behind developing ontologies of domains is defining data for other programs to use.

There are different ways to develop ontologies of domains/model domains. Ontology development is an iterative process which will continue during the lifecycle of the ontology.

Could you write a section that might be appended to this paper, Salah et al, 2016, which would present the next phase of evolution history, from microservices to the technologies which are commonly in use today?

As cloud-based technologies are becoming ubiquitous, more portability, efficiency and scalability are needed. To meet all of them, unikernels, AI operations and serverless frameworks are good help: Unikernels provide more portability and efficiency. They put necessary bit of code for running an application into a package. It makes it lighter and more secure. Artificial intelligence operations pave the way for more automation by building tools that automate the work. So, the problems can be targeted easily and fixed automatically. The serverless frameworks allow running code in more scalable and cost-effective way.