Cympha™

LANGUAGE that YIELDS MEDICAL PROCESS HANDLING

ALGORITHM

May 20, 2017

Rickard Verner Hultgren *rihu0003@student.umu.se*

A language would be useless if there is no way to understand it. The process of interpretation LYMPHA scripts works as follows:

1. *Structurizing*; All the objects are stored into a workig memory.
2. *Recording*; Linking the objects together.
3. *Playing*; The double linked list is gone through.

Each of these is coverd below.

1 Structurizing

The goal of the syntax is to build asignments for the two datatypes. These asignments will then be put into an list. Hence here are the node types for the linked list:

- Head node.
  - *node next*
- Events; Always end with a full stop (.).
  - *string name*
  - *list specifications*
  - *list content*
- Factors; Always end with a questionmark (?).
  - *string name*
  - *list specifications*
  - *list content*
  - *int tipping point*
  - *int relational operator*

2 recording

By the use of REGEX, data objects will be made. Each object has the capacity of pointing to multiple previous-node-objects as well as next-node-objects. Additionaly a list will be made, that stores pointers to all the objects.

```
# 1.1 Start a list
  p = ( head-of-object-list→next-object-list )
```

```
# 1.2 Make objects
  if REGEX(object found) then
    object = malloc(sizeof(struct node))
    ( p→next-object-list ) = object
    p = object
  end if
```

3 playing

Steps of searching object with the same name as in index from given start node.

EXE FUNCTION

```
  LIST-OF-OBJECTS-TO-EXECUTE
  for next-object in object.next-objects do
    for list-object in object-list do
      if list-object == next-object then
        pointer to list-object is added to LIST-OF-OBJECTS-TO-
        EXECUTE
      end if
    end for
  end for
  for exeobject in LIST-OF-OBJECTS-TO-EXECUTE do
    if exeobject.flow==1 then
      execute exeobject
      for subexeobject in exeobject.subobjects do
        execute subexeobject
      end for
      pointer to list-object is added to PAST-LIST-OF-OBJECTS-TO-
      EXECUTE
      delete exeboject-pointer in LIST-OF-OBJECTS-TO-EXECUTE
    end if
  end for
  for pastexeobject in PAST-LIST-OF-OBJECTS-TO-EXECUTE do
    EXECUTE FUNCITON (pastexeobect.next)
  end for
```

SHOW FUNCTION

```
LIST-OF-OBJECTS-TO-EXECUTE
for next-object in object.next-objects do
  for list-object in object-list do
    if list-object == next-object then
      pointer to list-object is added to LIST-OF-OBJECTS-TO-
      EXECUTE
    end if
  end for
end for
for exeobject in LIST-OF-OBJECTS-TO-EXECUTE do
  execute exeobject
  for subexeobject in exeobject.subobjects do
    show subexeobject
  end for
  pointer to list-object is added to PAST-LIST-OF-OBJECTS-TO-
  EXECUTE
  delete exeboject-pointer in LIST-OF-OBJECTS-TO-EXECUTE
end for
for pastexeobject in PAST-LIST-OF-OBJECTS-TO-EXECUTE do
  execute exeobject
end for
```

MAP FUNCTION

```
LIST-OF-OBJECTS-TO-EXECUTE
for next-object in object.next-objects do
  for list-object in object-list do
    if list-object == next-object then
      pointer to list-object is added to LIST-OF-OBJECTS-TO-
      EXECUTE
    end if
  end for
end for
for exeobject in LIST-OF-OBJECTS-TO-EXECUTE do
  if exeobject.flow==1 then
    execute exeobject
    for subexeobject in exeobject.subobjects do
      map subexeobject
    end for
  end if
  pointer to list-object is added to PAST-LIST-OF-OBJECTS-TO-
  EXECUTE
  delete exeboject-pointer in LIST-OF-OBJECTS-TO-EXECUTE
end for
for pastexeobject in PAST-LIST-OF-OBJECTS-TO-EXECUTE do
  execute exeobject
end for
```

4 Tutorial

The main goal is to calculate a proposal for the next step in a patientâĂŹs investigation. In future, a prioritization feature is planned. Before reading a script, one must decide the following things:

- Where to start reading. There are two alternatives:
  - Specify starting points for reading.
  - Use objects that has no previous object linked to them.
- How many steps should be read? The followig alternatives are available:
  - Specify how many steps should be read.
  - Read until the script ends.