Universidade Estadual de Feira de Santana - UEFS
Programa de Pós-Graduação em Computação Aplicada - PGCA

Nome do Aluno: ____

Prova para comprovação de proficiência em língua inglesa (2016.1)
Compreensão e interpretação de texto referente à literatura técnica ou científica em língua inglesa

1st Part: Multiple choice. Use only ball pen for answering the questions.

TEXT 1

Biology software promises easier way to program living cells

'Cello' automates the fast, reliable design of DNA-based logic circuits.

Erika Check Hayden

Nature News, doi:10.1038/nature.2016.19671

Synthetic biologists have created software that automates the design of DNA circuits for living cells. The aim is to help people who are not skilled biologists to quickly design working biological systems, says synthetic biologist Christopher Voigt at the Massachusetts Institute of Technology in Cambridge, who led the work. “This is the first example where we’ve literally created a programming language for cells,” he says.

In the new software — called Cello — a user first specifies the kind of cell they are using and what they want it to do: for example, sense metabolic conditions in the gut and produce a drug in response. They type in commands to explain how these inputs and outputs should be logically connected, using a computing language called Verilog that electrical engineers have long relied on to design silicon circuits. Finally, Cello translates this information to design a DNA sequence that, when put into a cell, will execute the demands.

Voigt says his team is writing user interfaces that would allow biologists to write a single program and be returned different DNA sequences for different organisms. Anyone can access Cello through a Web-based interface, or by downloading its open-source code from the online repository GitHub.

”This paper solves the problem of the automated design, construction and testing of logic circuits in living cells,” says bioengineer Herbert Sauro at the University of Washington in Seattle, who was not involved in the study. The work is published in Science.

1) Based on Text 1, choose the CORRECT answer: (0.8 point)
   a) Scientists have used electronic circuits in living cells.
   b) Living cells use DNA to automate software.
   c) A programming language was proposed for the design of DNA circuits for living cells.
   d) Only skilled biologists can design working biological systems.

2) According to Text 1, choose the WRONG answer: (0.8 point)
   a) To design a cell, the user specifies the desired input and output.
   b) Cello can sense the metabolic conditions in the gut and produce a drug.
   c) Cello uses input and output specifications to design a DNA sequence.
   d) Electrical engineers design silicon living cells.

3) According to Text 1, place right (R) or wrong (W) for each alternative. (0.9 point)

( ) a) Only electrical engineers can access Cello.
( ) b) Living cells are used in Cello to program DNA.
( ) c) Cello provides different DNA sequences for a single program.
( ) d) Cell can help design, construct and test logic circuits in living cells.

TEXT 2

Automating Science

David Waltz, Bruce G. Buchanan

Science 03 Apr 2009: Vol. 324, Issue 5923, pp. 43-44

The idea of automating aspects of scientific activity dates back to the roots of computer science, if not to Francis Bacon. Some of the earliest programs automated the processes of creating ballistic tables, cracking cyphers, collecting laboratory data, etc., by carrying out a set of instructions from start to finish. Starting with DENDRAL in the 1960s (1), artificial intelligence programs such as Prospector (2), Bacon (3), and Fahrenheit (4) automated some of the planning, analysis, and discovery portions of the scientific enterprise. However, most of these programs were still designed to run a calculation to completion, produce an answer, and then stop. They did not fully "close the loop" in the sense of examining the results of their actions, deciding what to try next, potentially cycling forever.

[...]

The main goals of automation in science have been to increase productivity by increasing efficiency (e.g., with rapid throughput), to improve quality (e.g., by reducing error), and to cope with scale, allowing scientific treatment of topics that were previously impossible to address. Tycho Brahe spent a lifetime recording observations that allowed Johannes Kepler to formulate Kepler's laws of planetary motion; today, computer-controlled data collection is commonplace and necessary for both experimental and observational science. Automating many activities beyond data collection offers even more benefits.

In the near term, a useful metaphor is to consider computers as intelligent assistants. Some assistants gather data and attend to such tasks as noise filtering, data smoothing, outlier rejection, and data storage. Other assistants are specialists at statistical analysis, still others at bench work. This metaphor has driven many research projects over the past several decades and has led to many of the most successful applications of computers.

4) Based on Text 2, choose the CORRECT answer: (0.8 point)

a) The roots of scientific activity is in computer science.
b) Kepler's laws were addressed by computers.
c) Computers can control data collection.
d) Scientific treatment of research tasks are impossible.

5) According to Text 2, choose the WRONG answer: (0.8 point)

a) Computers can act as intelligent assistants producing noise and outliers.
b) Programs from the 1960s were used to run a calculation, produce a result and stop.
c) Automation of science can improve the quality of scientific research.
d) Data smoothing and storage are some of the uses of computers in science automation.

6) According to Text 2, place right (R) or wrong (W) for each alternative. (0.9 point)

(   ) a) Bacon was used to automate Fahrenheit scientific research.

(   ) b) Ballistic tables were created by computer programs.

(   ) c) Computer programs never close the loop.

(   ) d) Computer programs automate science by examining results and deciding the next step.

2nd Part: Open Questions. Answer each question in <u>Portuguese</u>, using only the designated lines. Use only ball pen for answering the questions. All questions refer to the TEXT 3.

TEXT 3

Autonomous Mini Rally Car Teaches Itself to Powerslide

Evan Ackerman
IEEE Spectrum, 18 May 2016

Most autonomous vehicle control software is deliberately designed for well-constrained driving that's nice, calm, and under control. Not only is this a little bit boring, it's also potentially less safe: If your car autonomous vehicle has no experience driving aggressively, it won't know how to manage itself if something goes wrong.

At Georgia Tech, researchers are developing control algorithms that allow small-scale autonomous cars to power around dirt tracks at ludicrous speeds. They presented some this week at the 2016 IEEE International Conference on Robotics and Automation in Stockholm, Sweden. Using real-time onboard sensing and processing, the little cars maximize their speed while keeping themselves stable and under control. Mostly.

The electrically powered research platform, which is a scale model one-fifth the size of a vehicle meant for human occupants, is called AutoRally. It's about a meter long, weighs 21kg, and has a top speed of nearly 100 kilometers per hour. It's based on an R/C truck chassis, with some largely 3D-printed modifications to support a payload that includes a GPS, IMU, wheel encoders, a pair of fast video cameras, and a beefy quad-core i7 computer with a Nvidia GTX 750ti GPU and 32 gigs of RAM. All of this stuff is protected inside of an aluminum enclosure that makes crashing (even crashing badly) not that big of a deal.

[...] The real magic here is the algorithm that manages AutoRally’s steering and throttle. Rather than hierarchically splitting control and planning into two separate problems, Georgia Tech's algorithm combines them by integrating vehicle dynamics in real-time. Generally, this is a very computationally intensive approach, but AutoRally can calculate an optimized trajectory from the weighted average of 2,560 different trajectory possibilities, all simulated in parallel on to the monster onboard GPU. Each of these trajectories represents the oncoming 2.5 seconds of vehicle motion, and AutoRally recomputes this entire optimization process 60 times every second.

The initial training phase consists of just a few minutes of a non-expert human driving AutoRally around the track in remote-control mode; all of the fancy stuff (like the powersliding) is a product of the algorithm itself. Such aggressive driving is necessary when the speed of the vehicle exceeds its friction limit—a potentially dangerous condition for inexperienced robot drivers and human drivers alike. And this is why research into aggressive driving is not just fun but important. Just as expert human drivers can take advantage of how their vehicles handle at the very limits of control in order to drive fast yet safely in extremely challenging conditions, self-driving cars should be able to use the same techniques to avoid accidents in bad weather. [...]

Question 1: According to text 3, which are the main computer hardware features of AutoRally? (2 points)

Question 2: According to text 3, which are the differences and advantages of AutoRally's control algorithm when compared to other approaches? (1.5 point)

Question 3: According to text 3, why is aggressive driving relevant as a topic of research? (1.5 point)