

### Many kinds of exercise types in A+ not just programming 2nd A+ WorldCon 7.11.2018 Tampere, Finland

### Markku Riekkinen

Department of Computer Science Aalto University School of Science markku.riekkinen@aalto.fi

7.11.2018

### Contents

- Acos server
- Jsvee program visualizations
- js-parsons programming puzzles
- JSAV algorithm visualizations
- Point-and-click
- Drag-and-drop
- Databases: relational algebra and SQL queries
- Theoretical Computer Science: finite state automata, context-free grammars
- LTI: standard protocol for integrating content and exercises



### **Acos server**

- Platform for building typically *frontend-heavy exercises* (browser JavaScript). Less emphasis on backend computation.
- ► Acos supports multiple protocols (between LMS and Acos) and exercises are implemented independent of the protocol → interoperability with many learning systems
- https://github.com/acos-server/acos-server
- Public demos https://acos.cs.aalto.fi/



### **Acos server**



Content developers do not need to worry about supporting different protocols. Protocol developers can focus on supporting their protocol without concerns on particular content types. Instructors are free to use different content types and protocols as needed. Learners do not need to know about the services or protocols used. All content appears within the same LMS.

Figure: Teemu Sirkiä, https://github.com/acos-server/acos-server/blob/master/doc/acos\_concept.jpg



### Jsvee program visualizations

- Visualize the program execution inside the computer step-by-step
- Program counter, function call stack, variables, heap
- Supports Python, C#, Scala, Java
- Use via Acos or host static files somewhere
- https://github.com/acos-server/acos-jsvee
- https://github.com/Aalto-LeTech/jsvee
- Tutorial: https://www.youtube.com/watch?v=Q3T\_QLRWb78



## Jsvee program visualizations

Course	Points (110) My submissions (110) + Deadline Sat, Jun 6 2020, noon
<ul> <li>Course materials</li> <li>It Exercise results</li> <li>Course staff</li> <li>Participants</li> <li>Arresults</li> <li>Arresults</li> <li>Arresults</li> <li>It Visualizations</li> <li>✓ Edit news</li> <li>✓ Edit course</li> </ul>	<pre>bis example, we will see how functions work. In this case we will see how the parameter passing and the return value work. Please pay attention because functions are very important but sometimes have to understand. Can you understand the difference between printing and returning a value after this example. We will see how the parameter functions are very important but sometimes have the difference between printing and returning a value after this example. We will see how the parameter functions are very important but sometimes have the difference between printing and returning a value after this example. We we will see how the parameter functions are very important but sometimes functions are very important but sometimes functions are very important but sometimes for example. We will see how the parameter functions are very important but sometimes functions are very important but sometimes for example. We will see how the parameter functions are very important but sometimes for example. We we will see how the parameter functions are very important but sometimes for example. We were sometimes for example, we will see how the parameter for example, we will see how the parameter for example. The parameter for example, we will see how the parameter fo</pre>
	Fetching value from the variable height - ready.





## js-parsons programming puzzles

- Construct Python programs from given lines/blocks of code by dragging and dropping
- Use via Acos or host static files somewhere
- https://github.com/acos-server/acos-jsparsons
- https://github.com/acos-server/acos-jsparsons/ blob/master/static/js-parsons/README.md



## js-parsons programming puzzles

### js-parsons programming puzzles

Construct Python programs from the given blocks of code.

Points 0 / 10	My submissions 0/10 -	Dead	line Sat, Jun 6 2020, noon	View all submissions
Drag from here			Construct your solu	ution here
elif a [??] b:			<pre>def return_bigger_or_none(a,b):</pre>	
return a			if a [ b:	
return b			return None	
pass				

#### New instance Get feedback

Construct a function that returns the bigger value of the given arguments. If a and b are equal, it should return None.



# JSAV algorithm visualizations

- Manipulate a visual representation of a data structure to simulate the behaviour of the algorithm
- For example: sorting, trees, graphs, hash algorithms
- http://jsav.io/
- https://github.com/vkaravir/JSAV
- JSAV has not yet been properly integrated into Acos
- OpenDSA project: https://opendsa-server.cs.vt.edu/



# **JSAV** algorithm visualizations

#### Breadth-first search (BFS)





## **JSAV** algorithm visualizations





### **Point-and-click**

- The student should click on the correct targets out of all available clickables
- Immediate feedback given for each click
- Exercise defined with an HTML and a JSON file
- Originally designed for language studies, but could be used for other cases too
- The exercise content is not limited to just text as it is defined in HTML
- https://github.com/acos-server/acos-pointandclick
- https://github.com/acos-server/ acos-pointandclick-example



## **Point-and-click**



#### Given-new principle

Point and click the main verbs in each sentence.

<sup>1</sup>Creative Commons <u>provides</u> free, easy-to-use copyright licenses allowing a simple, standardized way to <u>give</u> the public permission to share and use creative work. <sup>2</sup>In addition, CC licenses [Let] users <u>leasily</u> change their copyright terms from the default of "all rights reserved." <sup>3</sup>Although Creative Commons licenses are not an alternative to copyright, they work alongside copyright and allow users to modify copyright terms to best suit their needs. <sup>4</sup>Currently, the CC Affiliate Network consists of 100+ affiliates working in over 79 jurisdictions to support and promote CC activities around the world.

Adapted from "What is Creative Commons?" available at <u>https://creativecommons.org/about/</u>. This work is licensed under a <u>Creative Commons Attribution 4.0 International License</u>.



#### 😌 Correct!

easily change is the second element in the main verb phrase in sentence 2. The whole verb phrase is "let ... change". The adverb "easily" describes the action indicated in the verb "change".

```
3 more correct answers left! Correct: 3 Wrong: 1
```

Submit unfinished



## **Drag-and-drop**

- The student should drag the correct alternative to each gap
- Immediate feedback given for each drag
- Exercise defined with an HTML and a JSON file
- Originally designed for language studies, but could be used for other cases too
- The exercise content is not limited to just text as it is defined in HTML
- https://github.com/acos-server/acos-draganddrop

### https:

//github.com/acos-server/acos-draganddrop-example





Points 0/10 My submissions 0/10 - Deadline Sat. Jun 6 2020, noon

 a A an An the The ø

 Add indefinite and definite articles into the text below by dragging. Drag the "empty set" to the gaps that should be left empty.

 In @ mathematics and @ computer science, @n algorithm is @ self-contained sequence of the actions to be performed. @ Algorithms perform \_\_\_\_\_\_\_ calculation, \_\_\_\_\_\_\_ data processing, and/or \_\_\_\_\_\_\_ automated reasoning tasks.

 algorithm is \_\_\_\_\_\_\_ effective method that can be expressed within \_\_\_\_\_\_\_ finite amount of \_\_\_\_\_\_\_ space and time [1] and in \_\_\_\_\_\_\_ well-defined formal language [2] for calculating \_\_\_\_\_\_\_ function. [3] Starting from \_\_\_\_\_\_\_\_ initial state and \_\_\_\_\_\_\_\_ initial input (perhaps empty), [4] \_\_\_\_\_\_\_\_ instructions describe \_\_\_\_\_\_\_\_\_

View all submissions

states, eventually producing "output"[6] and terminating at final ending state.
transition from one state to next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate in random input [7].

concept of \_\_\_\_\_algorithm has existed for centuries; however, \_\_\_\_\_partial formalization of what would become \_\_\_\_\_modern algorithm began with \_\_\_\_\_attempts to solve \_\_\_\_\_Entscheidungsproblem (the "decision problem") posed by David Hilbert in \_\_\_\_\_1928.

This text is adapted from <u>Algorithm - Wikipedia</u>. This text is licensed under the <u>Creative Commons Attribution-Share-Alike</u> License 3.0.

Correct! No article should be used here.

Correct: 5 Wrong: 1 Submit unfinished



### Databases: relational algebra and SQL queries

- Basic exercises about relational algebra and SQL queries
- "Write the expression of relational algebra / SQL query answering this question..."
- Minimalistic editor helps with entering mathematical symbols in expressions of relational algebra
- The grader compares the query output to the output of the model solution
- The course is not published openly in the Internet, thus the exercises are not directly available for everyone to use



# **Relational algebra**

### Problem 1

Consider the following relations.

```
Students(<u>ID</u>, name, program, year)
Courses(<u>code</u>, name, credits)
Grades(<u>studentID</u>, <u>courseCode</u>, date, grade)
```

The Students relation gives the student ID, the name, the study program and the starting year of a student. The Courses relation gives the course code, the name and the number of credits of a course. The Grades relation gives the ID of the student who has completed the course, the code of the completed course and the date and the grade. We assume that each student can complete a certain course only once and only information on accepted grades are stored in relation Grades.

Write an expression of relational algebra to answer the following query: find the code and name of all courses whose number of credits is over 6.

Instructions



 $\pi_{code,name}(\sigma_{credits \ < \ 6}(Courses))$ 

#### Answer in relational algebra

 $\pi_{code,name}(\sigma_{credits < 6}(Courses))$ 

#### Submit



### **Relational algebra: feedback**

Exercise description

My submissions (1/5) -

Model answer

Q Inspect submission

View all sub

Your submission was ran against a database and the result was compared with a model answer.

Submission	results	Model answer results	
code	name	code	name
CS-A1110	Programming 1	CSE- A0042	Artificial Intelligence
CSE-A1111	Basic Course in Programming Y1	CSE- A2090	Operating Systems
CSE-A1200	Databases	CSE- A3302	Software Engineering
GC-H1110	How To Become a Game Changer	CSE- A3963	Information Security



# **SQL** query

### Problem 1

Consider a database schema which consists of four relations, whose schemas are:

```
Product(model, maker, type)
PC(model, speed, ram, hd, price)
Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)
```

See Exercise Round 1 for detailed description of the relations and their attributes. The possible values for attribute type of relation **Product** are pc, laptop and printer. The possible values for attribute type of relation **Printer** are laser, ink-jet, and other. The possible values for attribute color are 1 for color printers and 0 for other printers.

Use SQLiteStudio to test your queries before you submit your solutions to A+. See the instructions and a test database in MyCourses->Harjoitukset->Harjoitukset->Harjoitukset->Exercises Round 2) for instructions and an example database.

Write the following query in SQL: Find the model number, hard-disk size and price of all laptops whose screen size is at least 15 inches.

#### Answer in SQL

SELECT model FROM Laptop WHERE screen = 15;

#### Submit



## **Theoretical Computer Science**

- Basic exercises about formal languages, finite state automata, regular expressions, and context-free grammars
- Many variations of the exercise types: for example, give DFA for the language, from NFA to DFA, minimize DFA
- Multiple instances of each exercise: the student receives one randomly (i.e., the language is randomly selected from a pre-defined set of languages)
- The course is not published openly in the Internet, thus the exercises are not directly available for everyone to use



## **Theoretical Computer Science: formal languages**

### Words in a language

Consider the language  $L = \{w \in \{a, b\} * \mid w \text{ contains at least two } a's \}.$ 

Give 5 distinct words that belong to the language.

Each word must be of length at most 10.

You can use  $\epsilon$  or the underline symbol \_ to denote the empty word.

Your answer:

• Word 1:	aa
• Word 2:	aba
• Word 3:	baa
• Word 4:	a
• Word 5:	b
Submit!	



# **Theoretical Computer Science: deterministic finite**

### automata

Designing a DFA for a language

Consider the language  $L = \{w \in \{a, b\}^* \mid w \text{ contains the substring } aa \text{ or } bb \text{ (or both)}\}.$ 

Design a deterministic finite state automaton (DFA) that accepts the language.





### Theoretical Computer Science: DFA feedback Points 0/1

Your task was to give a deterministic automaton accepting the language  $L = \{w \in \{a, b\}^* \mid w \text{ contains the substring } aa \text{ or } bb \text{ (or both)}\}.$ 

Your solution is:



Your automaton does not accept the word 'aab' that belongs to the language.



# Theoretical Computer Science: regular expressions

### Regular expressions for a language

Consider the language  $L = \{w \in \{a, b\}^* \mid w \text{ contains the substring } aa \text{ or } bb \text{ (or both)}\}.$ 

Give a regular expression describing the language.

The syntax for regular expressions is:

- alphabet: a, b, ..., z, 0, 1, ..., 9
- parentheses: (, )
- union: | (the vertical stroke symbol)
- empty string: \_ (the underscore symbol) or ε (the greek epsilon)
- empty set: @ (the 'at'-symbol) or Ø (AltGr + Shift + ö on some keyboards)
- Kleene star: \* (the asterisk symbol)

For example, the regular expression \_laa(bc)\* describes the language  $\{\varepsilon, aa, aabc, aabcbc, aabcbcc, \dots\}$  over the alphabet  $\{a, b, c\}$ .

Your solution: a\*(aa|bb)b\*

Submit!



### Theoretical Computer Science: parse tree in CFG

Parse tree for a string in a grammar

Consider the context-free grammar:

S -> ASB | 0 A -> 0 | 1 B -> 0 | 1 | ε

Give a parse tree for the word "001" in the grammar.

Give your answer here. Click on parse tree nodes to expand and collapse them.





# **Theoretical Computer Science: context-free**

### grammars

Grammar for a language

Consider the language  $L = \{uvw \mid u, v, w \in \{0, 1\}^* \text{ and } v = w^R\}.$ 

Give a context-free grammar describing the language.

Enter the grammar in the space below, one production on each line. Use this syntax:

- production: nonterminal -> string1 | string2 ... | stringn
- nonterminals: capital letters A,B,...,Z
- terminals: lower case letters a,b,...,z and numbers 0,1,...,9
- empty string: \_ (the underscore symbol)
- · start symbol: the first nonterminal in the grammar

```
For instance, the grammar
```

```
A -> aAa | B
B -> bB |
```

describes the language  $\{a^n b^k a^n \mid n \ge 0 \text{ and } k \ge 0\}.$ 

Your solution:	
S-> UV	
U -> 0   1	
V -> 1V0	1
Submit!	



# Learning Tools Interoperability (LTI)

- Standard protocol for integrating content and exercises to learning management systems (LMS)
- Widely used in the world and supported by popular LMSes (Moodle, Canvas, etc.)
- A+ supports LTI 1.1
- LTI is more restricted as a protocol than the A+ exercise protocol
- Since all the major LMSes support LTI, there are plenty of LTI Tool Providers available that sell their content for a license fee



# LTI in A+: content embedded in an iframe



MATLAB Grader LTI Tool Provider: https://se.mathworks.com/products/matlab-grader.html

