HW1

Timo Wang

January 22, 2020

1 Objects

1.1 Courses

Courses are represented as follows:

a `cs111` - CS 111

b `cs211` - CS 211

c `cs321` - CS 321

d `cs330` - CS 330

e `cs335` - CS 335

f `cs338` - CS 338

g `cs348` - CS 348

h `cs371` - CS 371

1.2 Students

Some students are also listed here:

a `xyz123` - student XYZ123

b `xyz321` - student XYZ321

c `zyx123` - student ZYX123

2 Types and attributes

2.1 Types

The name of the type ”CS course” is `CSCourse`. The courses and their corresponding types are represented as follows:

a `CSCourse(cs111)`

b `CSCourse(cs211)`

c `CSCourse(cs348)`

d `CSCourse(cs371)`

A more general type would be ”Course”, named as `Course` and `isa(CSCourse, Course)` Furthermore, `isa(Course, Object)` where `Object` is the ”root” type.

The name of the type ”student” is `Student`. Some students and their corresponding types are represented as follows:

a `Student(xyz123)`

b `Student(xyz321)`

c `Student(zyx123)`

2.2 Attributes

I will first declare the followings:

a `IsAICourse(CSCourse)` - A CS course is an AI course

b `IsSystemCourse(CSCourse)` - A CS course is a system course

c `IsTheoryCourse(CSCourse)` - A CS course is a theory course

d `IsInterfaceCourse(CSCourse)` - A CS course is an interface course

e `IsSoftwareDevCourse(CSCourse)` - A CS course is a software development course

The listed courses are represented as follows as having the specified attributes:

a AI

- `IsAICourse(cs348)`
- `IsAICourse(cs371)`

b System

- `IsSystemCourse(cs321)`

c Theory

- `IsTheoryCourse(cs335)`

d Interface

- `IsInterfaceCourse(cs330)`

e Software development

- `IsSoftwareDevCourse(cs338)`

3 Relations

The ”pass” relationship between a student and a course is named `Pass` and defined as `Pass(Student, CSCourse)`. This relation has an arity of 2. Some example usages of it are listed below:

a `Pass(xyz123, cs371)` - Student XYZ123 passes CS371

b ¬`Pass(xyz123, cs348)` - Student XYZ123 does not pass CS348

c `Pass(xyz321, cs348)` - Student XYZ321 passes CS348

4 Functions

The function to represent the number of credits for a course is named numCreditOf and is defined to be numCreditOf(Course). It has an arity of 1.

A predicate Equal(Object, Object) is also defined to signify that two Objects are equal to each other.

The demonstrations of the function are then listed as follows:

a Equal(numCreditOf(cs371), 1)

b Equal(numCreditOf(cs371), numCreditOf(cs348))

c ∀ x [CSCourse(x) ⊃ Equal(numCreditOf(course), 1)]

5 Complex sentences

First, all new representations are defined. Then, the complex sentences are defined.

A predicate GreaterEqual(Object, Object) is defined to signify that the first Object, when applicable, is larger or equal to the second Object.

A function to represent the number of credits a student has earned is name numCSCreditEarnedBy and is defined to be numCSCreditEarnedBy(Student) with an arity of 1.

An attribute for a CS course to indicate whether it is a technical elective, IsTechnicalElective(CSCourse)

5.1 Problem 1

MeetCreditRequirement(s) ≡ ∃ x1, x2, ..., x16 [CSCourse(x1) ∧ CSCourse(x2) ∧ ... ∧ CSCourse(x16) ∧ Equal(numCreditOf(x1), 1) ∧ Equal(numCreditOf(x2), 1) ∧ ... ∧ Equal(numCreditOf(x16), 1) ∧ Pass(s, x1) ∧ Pass(s, x2) ∧ ... ∧ Pass(s, x16)]

5.2 Problem 2

MeetBreadthRequirement(s) ≡ ∃ x [CSCourse(x) ∧ IsAICourse(x) ∧ Pass(s, x)] ∧ ∃ x [CSCourse(x) ∧ IsSystemCourse(x) ∧ Pass(s, x)] ∧ ∃ x [CSCourse(x) ∧ IsInterfaceCourse(x) ∧ Pass(s, x)] ∧ ∃ x [CSCourse(x) ∧ IsSoftwareDevCourse(x) ∧ Pass(s, x)]

5.3 Problem 3

MeetDepthRequirement(s) ≡ ∃ x1, x2, x3, x4, x5, x6 [¬ (x1 = x2 ∨ x1 = x3 ∨ x1 = x4 ∨ x1 = x5 ∨ x1 = x6 ∨ x2 = x3 ∨ x2 = x4 ∨ x2 = x5 ∨ x2 = x6 ∨ x3 = x4 ∨ x3 = x5 ∨ x3 = x6 ∨ x4 = x5 ∨ x4 = x6 ∨ x5 = x6) ∧ CSCourse(x1) ∧ IsTechnicalElective(x1) ∧ CSCourse(x2) ∧ IsTechnicalElective(x2) ∧ CSCourse(x3) ∧ IsTechnicalElective(x3) ∧ CSCourse(x4) ∧ IsTechnicalElective(x4) ∧ CSCourse(x5) ∧ IsTechnicalElective(x5) ∧ CSCourse(x6) ∧ IsTechnicalElective(x6) ∧ Pass(s, x1) ∧ Pass(s, x2) ∧ Pass(s, x3) ∧ Pass(s, x4) ∧ Pass(s, x5) ∧ Pass(s, x6)]

5.4 Problem 4

MeetAllRequirements(s) ≡ MeetCreditRequirement(s) ∧ MeetBreadthRequirement(s) ∧ MeetDepthRequirement(s)