UNIVERSITÀ DI TRENTO

Computational Logic Exercises
Module V – LOD applications and LODE

Vincenzo Maltese

Well formed formulas

Which of the following complex formulas are syntactically correct in LOD? (TBOX formulas)

a) $A \equiv \exists R.C \sqcap \forall S.D$

b) $A \sqcap B \equiv C \sqcup D$

c) $A \equiv B \sqcap \neg C$

d) $A \sqsubseteq \neg C$

e) $A \sqsubseteq B \sqcap \exists R.C$

f) $A \sqsubseteq B \sqcap \exists R.(\forall S.D)$

g) $A \equiv B \sqcup \emptyset$

ANSWER:

a, c, d, e, f

UNIVERSITÀ DI TRENTO

Unfolding a Concept

Unfold ColouredGuitar:

$\text{ElectricGuitar} \equiv \text{Guitar} \sqcap \forall \text{hasSoundAmplification.withInputJack}$

$\text{ColouredGuitar} \equiv \text{ElectricGuitar} \sqcap \exists \text{hasColour.String}$

Answer:

$\text{ColouredGuitar} \equiv \text{Guitar} \sqcap \forall \text{hasSoundAmplification.withInputJack} \sqcap \exists \text{hasColour.String}$

Cyclic and acyclic TBOX

Is the following TBOX cyclic?

Woman $\equiv$ Person $\sqcap$ Female

Man $\equiv$ Person $\sqcap$ $\neg$Woman

Mother $\equiv$ Woman $\sqcap$ $\exists$hasChild.Person

Father $\equiv$ Man $\sqcap$ $\exists$hasChild.Person

Parent $\equiv$ Father $\sqcup$ Mother

ANSWER:

No, because by unfolding all concepts I never obtain the same concept on the left and on the right of the equivalences.

UNIVERSITÀ DI TRENTO

Cyclic and acyclic TBOX

Is the following TBOX cyclic?

Male ≡ ¬Female

Female ≡ ¬Male

ANSWER:

Yes, because by unfolding it I get Female ≡ ¬(¬Female) that is Female ≡ Female

Terminology

Is the following TBOX a terminology?

Mother $\equiv$ Woman $\sqcap$ $\exists$hasChild.Person

Father $\equiv$ Man $\sqcap$ $\exists$hasChild.Person

Parent $\equiv$ Father $\sqcup$ Mother

ANSWER:

Yes, because it is acyclic and there are only equivalences.

Terminology

Is the following TBOX a terminology?

$Mother \equiv Woman \sqcap \exists hasChild.Person$

$Father \equiv Man \sqcap \exists hasChild.Person$

$Parent \sqsubseteq Father \sqcup Mother$

ANSWER:
No, because it contains a subsumption.

Creating a terminology by expansion

Given the previous TBOX, provided below, can I convert it to make it a terminology?

Mother ≡ Woman ⊓ ∃hasChild.Person

Father ≡ Man ⊓ ∃hasChild.Person

Parent ⊑ Father ⊔ Mother

ANSWER:

Yes, for instance as follows:

Mother ≡ Woman ⊓ ∃hasChild.Person

Father ≡ Man ⊓ ∃hasChild.Person

StepMother ≡ Woman ⊓ ∃marriedWith.Father

StepFather ≡ Man ⊓ ∃marriedWith.Mother

Parent ≡ Father ⊔ Mother ⊔ StepFather ⊔ StepMother

Defining a terminology from natural language definitions

A lion is a large gregarious predatory feline of Africa and India having a shaggy mane in the male

$Lion \equiv Feline \sqcap Large \sqcap Gregarious \sqcap Predatory \sqcap \forall livesIn.(Africa \sqcup India) \sqcap \exists livesIn.(Africa \sqcup India)$

$MaleLion \equiv Lion \sqcap Male \sqcap \forall has.ShaggyMane \sqcap \exists has.ShaggyMane$

A penguin is a flightless bird of Antarctica having webbed feet

$Penguin \equiv Bird \sqcap \neg Fly \sqcap \forall livesIn.Antarctica \sqcap \exists livesIn.Antarctica \sqcap \forall has.WebbedFeet \sqcap \exists has.WebbedFeet$

Defining a terminology from a schema

Thing > Event

Property	Expected Type
Properties from Event	
about	Thing
actor	Person
attendee	Organization or Person

ANSWER. By assuming the schema as complete (otherwise it is not a terminology) we have:

$$\begin{aligned}\text{Event} \equiv \text{Thing} \sqcap \\ &\forall \text{about.Thing} \sqcap \exists \text{about.Thing} \sqcap \\ &\forall \text{actor.Person} \sqcap \exists \text{actor.Person} \sqcap \\ &\forall \text{attendee.(Person} \sqcup \text{Organization)} \sqcap \exists \text{attendee.(Person} \sqcup \text{Organization)}\end{aligned}$$

Logical consequences of a terminology (by unfolding)

Given the previous TBOX, replicated on the right, which of the following are logical consequences of the TBOX?

a) $Event \sqsubseteq \forall about.Thing \sqcap \exists about.Thing$

b) $Event \sqsubseteq \forall about.Thing$

c) $Event \sqsubseteq \exists about.Thing$

d) $Event \equiv \forall about.Thing \sqcap \exists about.Thing$

e) $Event \sqsubseteq \forall attendee.Person \sqcup \forall attendee.Organization$

f) $Event \sqsubseteq \forall attendee.Person$

g) $Person \sqsubseteq \neg Organization$

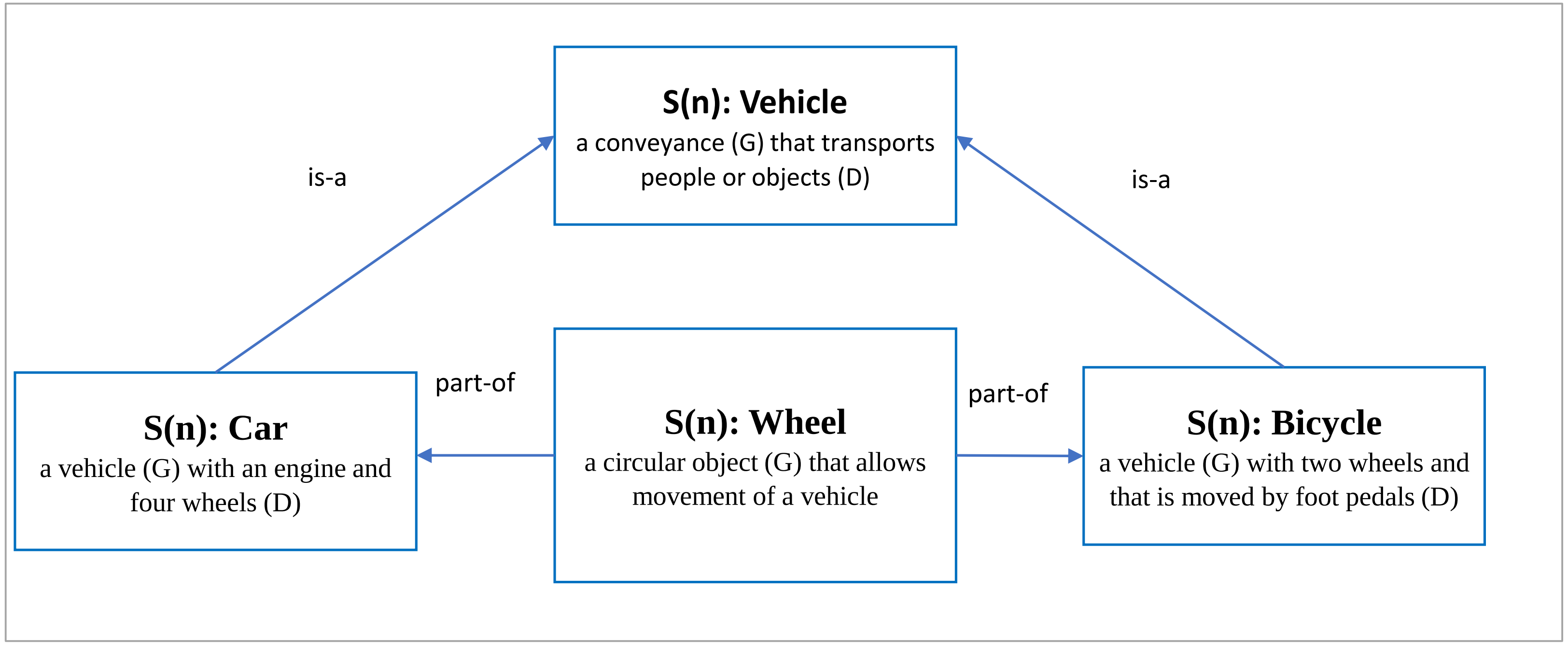
ANSWER.

a, b, c, f

TBOX

$Event \equiv Thing$
$\sqcap \forall about.Thing \sqcap \exists about.Thing$
$\sqcap \forall actor.Person \sqcap \exists actor.Person \sqcap$
$\forall attendee.(Person \sqcup Organization) \sqcap$
$\exists attendee.(Person \sqcup Organization)$

Formalizing a lexicon as a terminology (I)



Vehicle ≡ Conveyance ⊓ ∃transports.(Person ⊔ Object) ⊓ ∀transports.(Person ⊔ Object)

Car ≡ Vehicle ⊓ ¬Bicycle ⊓ ∃hasPart.Wheel ⊓ ∃hasPart.Engine

Bicycle ≡ Vehicle ⊓ ∃hasPart.Wheel ⊓ ∃hasPart.FootPedal ⊓ ∃movedBy.FootPedal ⊓ ∀movedBy.FootPedal

Wheel ≡ Object ⊓ CircularShape ⊓ ∃moves.Vehicle ⊓ ∀moves.Vehicle

Reasoning in LOD

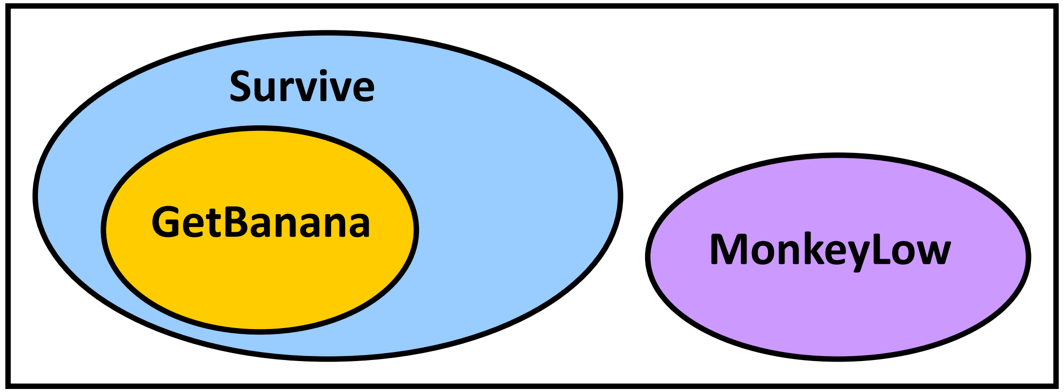
Suppose we model the Monkey-Banana problem as follows:

“If the monkey is low in position then it cannot get the banana. If the monkey gets the banana it survives”.

Theory T:
MonkeyLow $\sqsubseteq \neg$GetBanana
GetBanana $\sqsubseteq$ Survive

Is T satisfiasble?

ANSWER: Yes. It is enough to find one model for it, represented graphically with the Venn Diagram below.



Reasoning in LOD

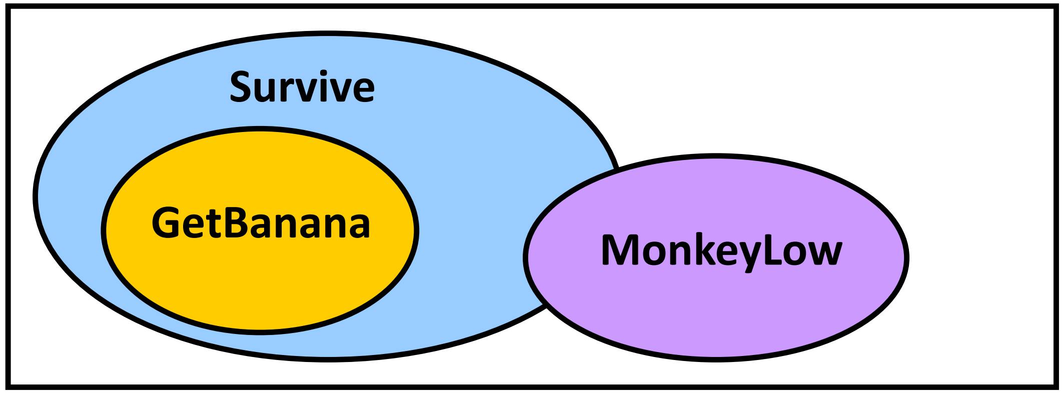
Suppose we model the Monkey-Banana problem as follows:

"If the monkey is low in position then it cannot get the banana. If the monkey gets the banana it survives".

Theory T:
MonkeyLow ⊑ ¬GetBanana
GetBanana ⊑ Survive

Is it possible for a monkey to survive even if it does not get the banana?

ANSWER: We can restate the problem as follow:
does T ⊨ ¬GetBanana ⊓ Survive at least in one model?
Yes. We can find a model in which both all the assertions in T and ¬GetBanana ⊓ Survive are not empty.



Reasoning in LOD

Suppose we describe students in a course as follows:

Undergraduate	$\sqsubseteq \neg$ Teach
Bachelor	$\equiv$ Student $\sqcap$ Undergraduate
Master	$\equiv$ Student $\sqcap \neg$ Undergraduate
PhD	$\equiv$ Master $\sqcap$ Research
Assistant	$\equiv$ PhD $\sqcap$ Teach

Are all assistants also undergraduates?

ANSWER: We can restate the problem as follow:
does T $\vDash$ Assistant $\sqsubseteq$ Undergraduate ?

We need to prove that this is true in <u>all models</u> (via the method of *unfolding*)

Assistant $\equiv$ PhD $\sqcap$ Teach
$\equiv$ Master $\sqcap$ Research $\sqcap$ Teach
$\equiv$ Student $\sqcap \neg$ Undergraduate $\sqcap$ Research $\sqcap$ Teach
Answer is No. Assistants are actually students who are not undergraduate.

Reasoning in LOD

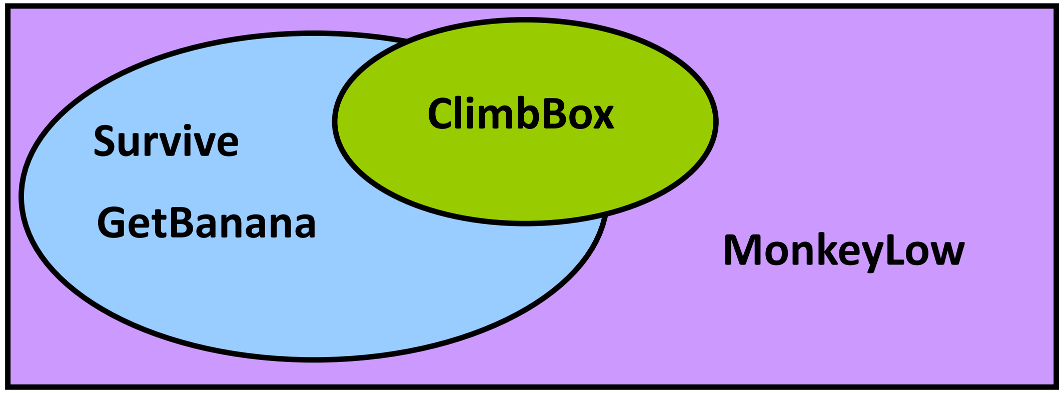
Suppose we model the Monkey-Banana problem as follows:

"If the monkey is low in position then it cannot get the banana. If the monkey gets the banana it survives".

Theory T:
MonkeyLow ≡ ¬GetBanana ⊓ ¬ClimbBox
GetBanana ≡ Survive

Is it possible for a monkey to climb the box and not survive?

ANSWER: We can restate the problem as follow:
does T ⊨ ClimbBox ⊓ ¬Survive at least in one model?
Yes. We can find <u>a model</u> in which both all the assertions in T and ClimbBox ⊓ ¬Survive are not empty.



Reasoning in LOD

Suppose we describe students in a course as follows:

Undergraduate	$\sqsubseteq \neg$ Teach
Bachelor	$\equiv$ Student $\sqcap$ Undergraduate
Master	$\equiv$ Student $\sqcap \neg$ Undergraduate
PhD	$\equiv$ Master $\sqcap$ Research
Assistant	$\equiv$ PhD $\sqcap$ Teach

Are bachelor and master disjoint?

ANSWER: We can restate the problem as follow:
does $T \vDash$ Bachelor $\sqcap$ Master $\sqsubseteq \bot$?

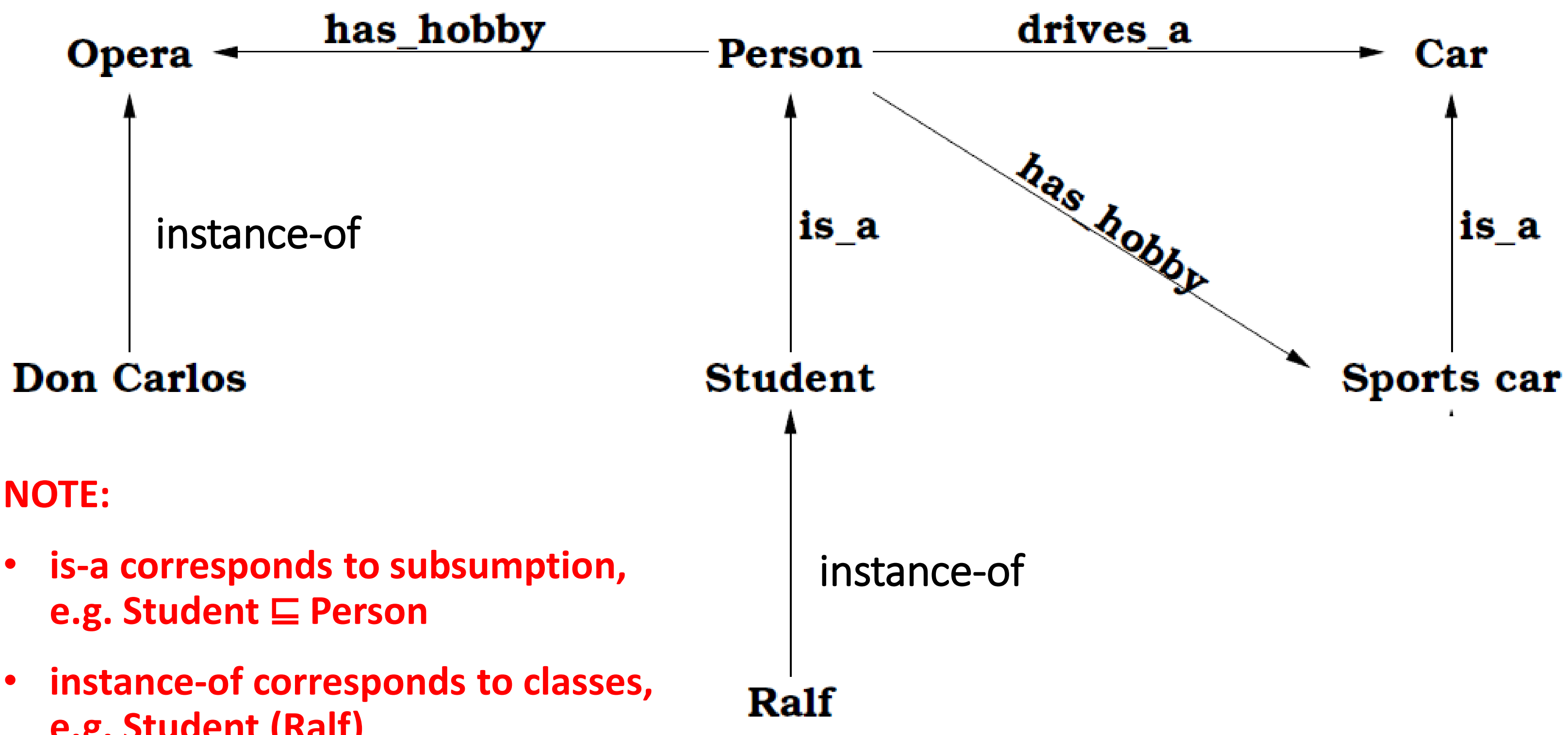
We need to prove that this is true in <u>all models</u> (via the method of *unfolding*)

Answer is obviously Yes because they contain two opposite constraints.

UNIVERSITÀ DI TRENTO

Define a LODE theory

Define a LODE theory for the following knowledge graph



NOTE:

- **is-a corresponds to subsumption, e.g. Student $\sqsubseteq$ Person**
- **instance-of corresponds to classes, e.g. Student (Ralf)**

ANSWER:

$$\text{Person} \sqsubseteq \exists\text{Drives.Car} \sqcap \exists\text{HasHobby.SportCar} \sqcap \exists\text{HasHobby.Opera}$$

$$\text{Student} \sqsubseteq \text{Person}$$

$$\text{SportCar} \sqsubseteq \text{Car}$$

Student(Ralf)

Opera(DonCarlos)

Define a LODE theory

Define a LODE theory for the following problem:

In a hospital patients, doctors and computers are equipped with proximity sensors able to detect whether doctors curated a patient or worked at their computer. The system detected that doctor Peter curated the patient Smith.

ANSWER:

$\text{Doctor} \sqsubseteq \forall \text{cure.Patient} \sqcap \forall \text{work.Computer}$

$\text{cure} \sqsubseteq \text{detected}$

$\text{work} \sqsubseteq \text{detected}$

Doctor (Peter) Patient (Smith) cure(Peter, Smith)

Expansion of a LODE concept

Given the following TBOX, compute the expansion of the ABox A = {StepMother(Mary)}

$Mother \equiv Woman \sqcap \exists hasChild.Person$

$Father \equiv Man \sqcap \exists hasChild.Person$

$StepMother \equiv Woman \sqcap \exists marriedWith.Father$

$StepFather \equiv Man \sqcap \exists marriedWith.Mother$

$Parent \equiv Father \sqcup Mother \sqcup StepFather \sqcup StepMother$

ANSWER:

StepMother(Mary), Woman(Mary), marriedWith(Mary, a1), Father(a1)

Man(a1), hasChild(a1, a2), Person(a2)

Expansion of a LODE concept

Given the following TBOX, compute the expansion of the ABox A = {StepMother(Mary), marriedWith(Mary, Paul)}

Mother ≡ Woman ⊓ ∃hasChild.Person
Father ≡ Man ⊓ ∃hasChild.Person
StepMother ≡ Woman ⊓ ∀marriedWith.Father
StepFather ≡ Man ⊓ ∃marriedWith.Mother
Parent ≡ Father ⊔ Mother ⊔ StepFather ⊔ StepMother

ANSWER:

StepMother(Mary), Woman(Mary), marriedWith(Mary, Paul), Father(Paul)
Man(Paul), hasChild(Paul, a1), Person(a1)

UNIVERSITÀ DI TRENTO

Instance checking in LODE

Given the following LODE theory T, does T |= Professor(John)?

Lecturer ≡ ∀Teaches.Course ⊓ ¬Undergrad ⊓ Professor

Lecturer (John)

Teaches(John, Logics)

Course(Logics)

ANSWER:

The expansion of Lecturer (John) is {Teaches(John, Logics), Course(Logics), ¬Undergrad(John), Professor(John)}

Therefore the answer is **yes**.

Instance retrieval in LODE

Given the following LODE theory T, find all the instances of Lecturer.

Lecturer ≡ ∀Teaches.Course ⊓ ¬Undergrad ⊓ Professor

Lecturer (John)

Teaches(John, Logics)

Course(Logics)

Teaches(Paul, Logics)

¬Undergrad(Paul)

Professor(Paul)

ANSWER:

{John, Paul}

In fact, John is in the ABox, while Paul satisfies all the constraints in the definition of Lecturer.

Concept realization in LODE

Given the following LODE theory T, find the most specific concept for Paul.

Lecturer ≡ ∀Teaches.Course ⊓ ¬Undergrad ⊓ Professor

Lecturer (John)

Teaches(John, Logics)

Course(Logics)

Teaches(Paul, Logics)

¬Undergrad(Paul)

Professor(Paul)

ANSWER:

Given that Paul satisfies all the constraints in the definition of Lecturer, the answer is Lecturer.

Note that if we remove Professor(Paul), the answer becomes {¬Undergrad}