Efficient Regression Testing of Ontology-Driven Systems

Mijung Kim,[*] Jake Cobb,[*] Mary Jean Harrold,[*] Tahsin Kurc,[†] Alex Orso,[*] Joel Saltz,[†] Andrew Post,[†] Kunal Malhota,[*] and Sham Navathe[*]

[*]Georgia Institute of Technology

[†]Emory University

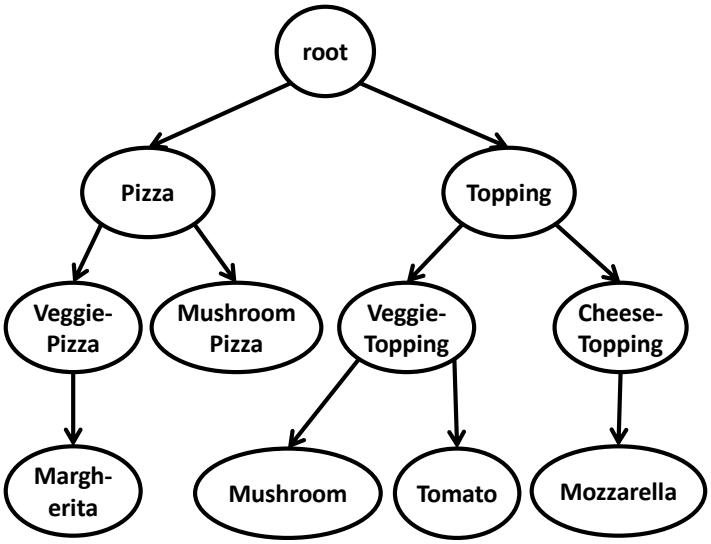
Supported by NSF CCF-0541048, CCF-0725202, CCF-0964647, CCF-1116210, NHLBI R24HL085343, NIH RC4MD005964, NIH/CTSA PHS UL1RR025008, IBM, and Microsoft

Ontologies

- An ontology is a **formal description of a domain** in terms of
  - Concepts within that domain
  - Relationships to each other
- Ontologies are used in, for example,
  - Biomedical
  - Dbpedia (wikipedia)
  - Finance
  - Music

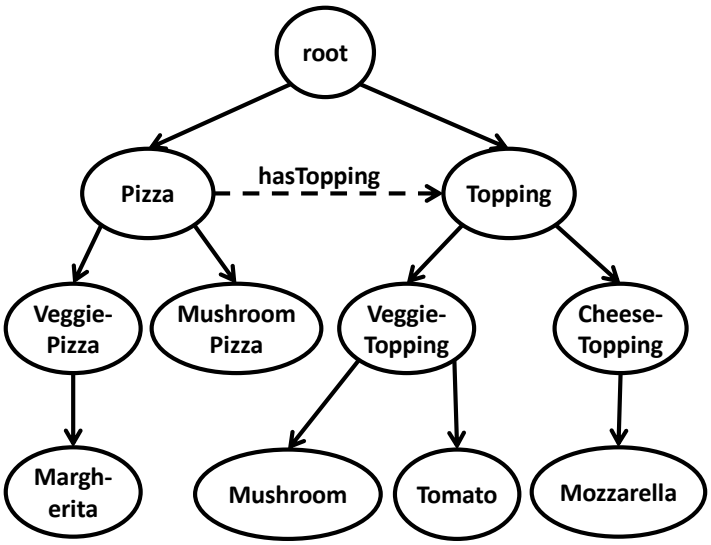
Components of an Ontology

- **Class**
  concept of a domain



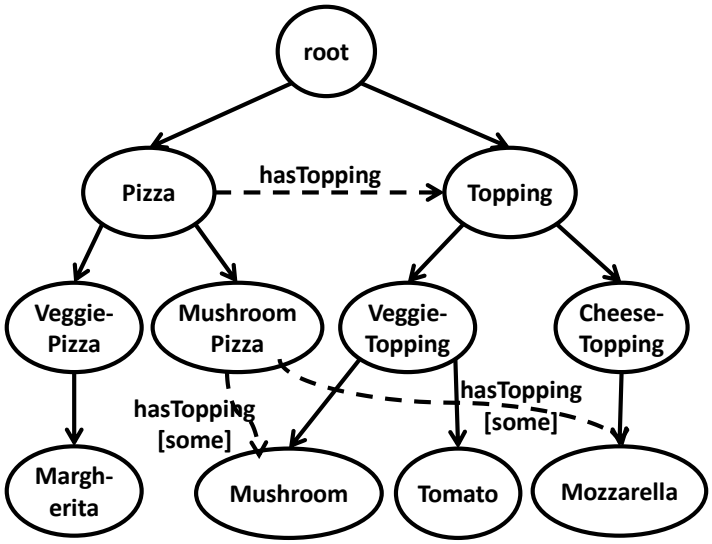
Components of an Ontology

- **Class**
  concept of a domain
- **Property**
  relationship between classes



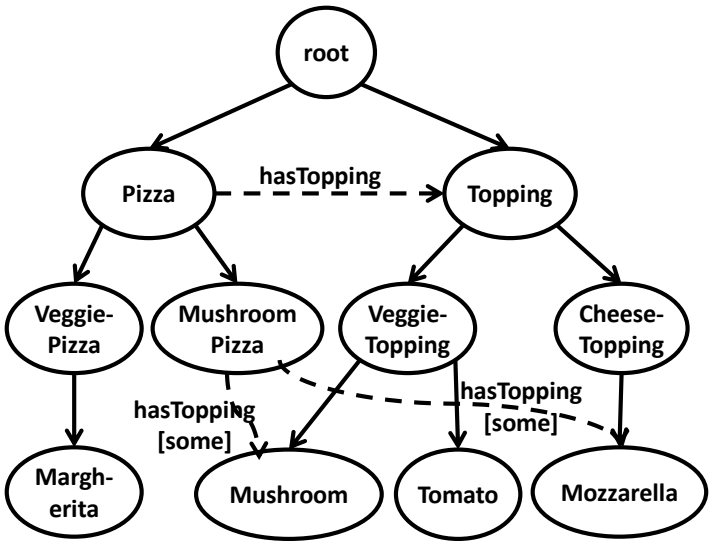
Components of an Ontology

- **Class**
  concept of a domain
- **Property**
  relationship between classes
- **Restriction**
  constraint on a class that indicates how a property is used



Benefits of an Ontology

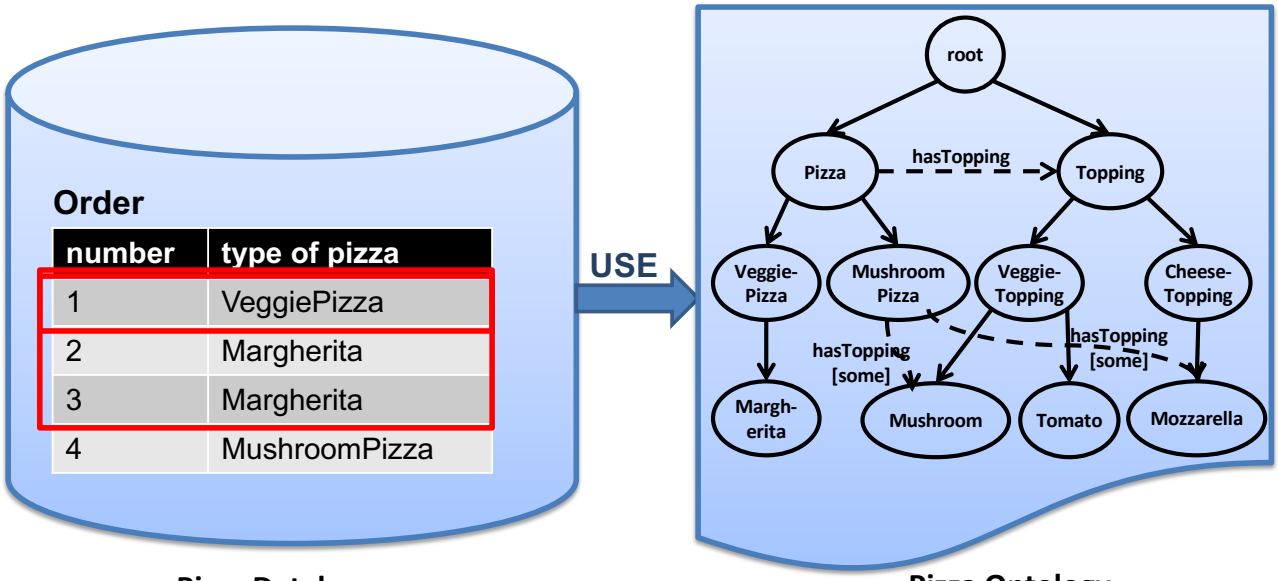
- Unifies structure of information
- Aggregates and correlates data to share information



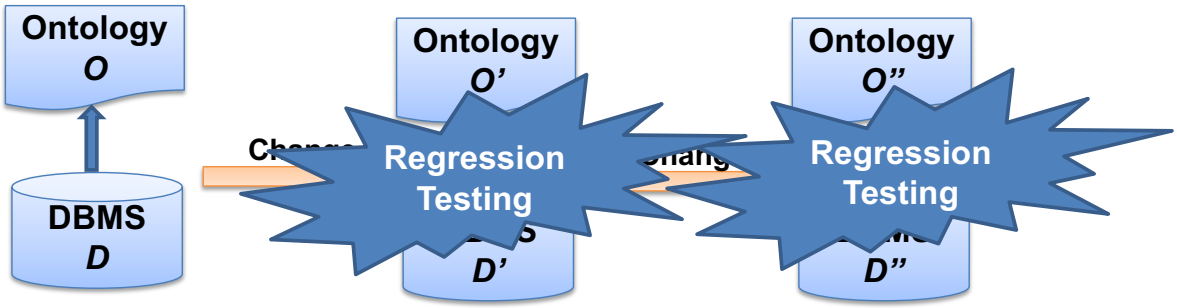
Ontology-Driven Systems

Query for "VeggiePizza" instances `from Order` returns

- **Database without an ontology: number 1 only**
- **Ontology-Driven DB: numbers 1, 2, and 3**



Regression Test Selection (RTS)



- Existing RTS techniques [White et al. ICSM'92, Chen et al. ICSE'94, Rothermel et al. TSE'97, Harrold et al. OOPSLA'01, Orso et al. FSE'04, Wilmor et al. ICSM'05, Haraty et al. SAC'01 and Nanda et al. ICST'11]

  Our technique supports ontology-driven DB systems

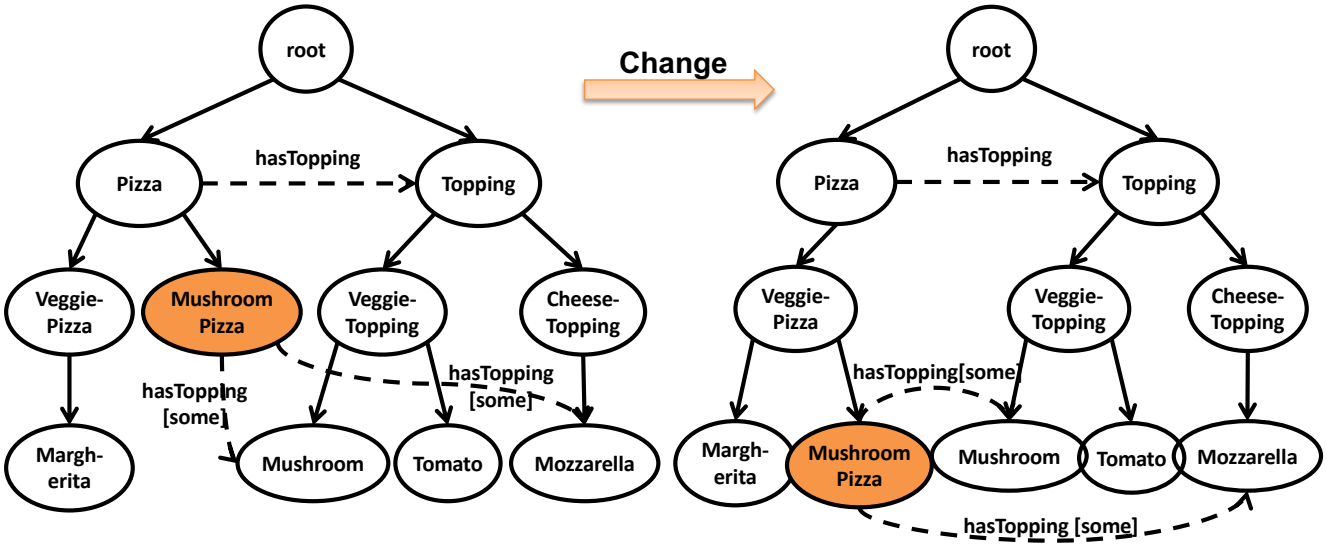
- Existing ontology differencing techniques [Noy et al. ISWC'04, Klein et al. EKAW'02]

  Our technique supports
  - **RTS in terms of changes in ontologies**
  - **Safety and scalability in detecting changes**

Our Work

- RTS technique for ontology-driven systems
  - Given O and O', identifies affected ontology entities
  - Selects test cases associated with affected entities
- Empirical studies with our tool, OntoReTest
  - Performed on two real systems in biomedical domain
  - Show effectiveness and efficiency

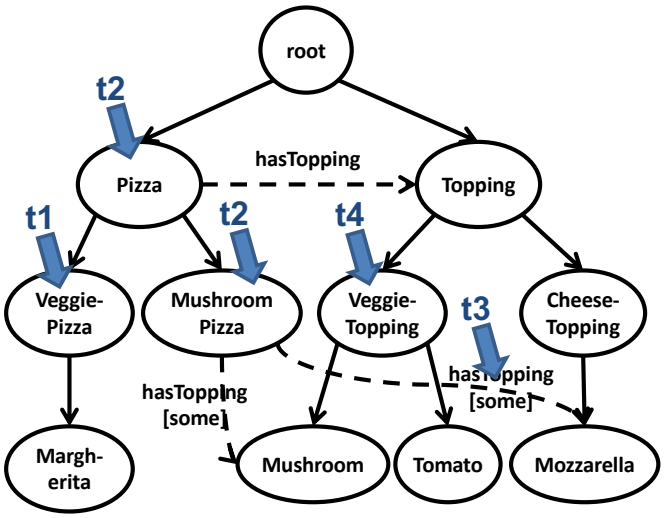
Example



G (original ontology)

G' (changed ontology) with 1 change

Example

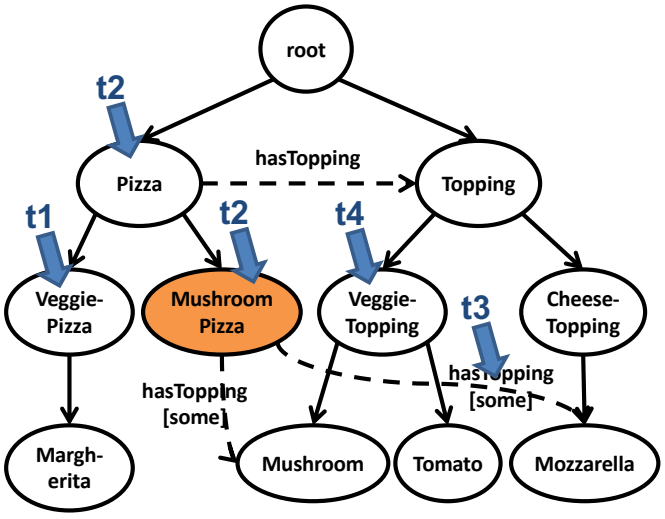


Test Case		Entities
t1	There are at least 3 veggie pizzas	<VeggiePizza>
t2	Pizza must include mushroom pizza	<Pizza>, <MushroomPizza>
t3	Mushroom pizza must have mozzarella toppings	<MushroomPizza hasTopping [some] Mozzarella>
t4	There must be at least 1 veggie topping	<VeggieTopping>

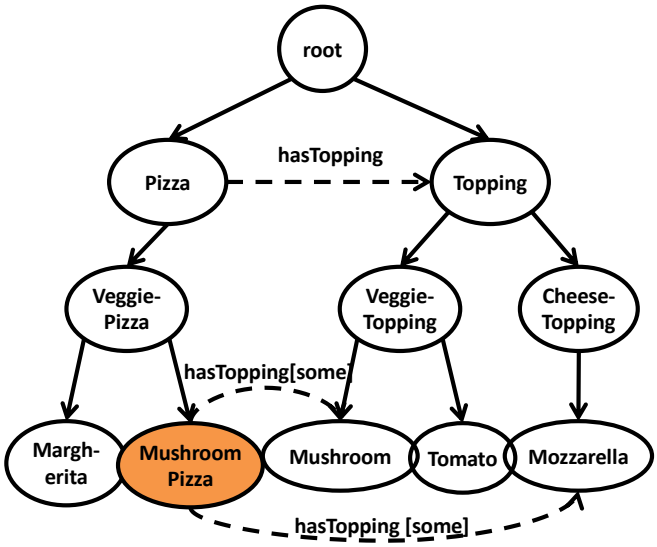
G (original ontology)

Coverage Matrix

Example

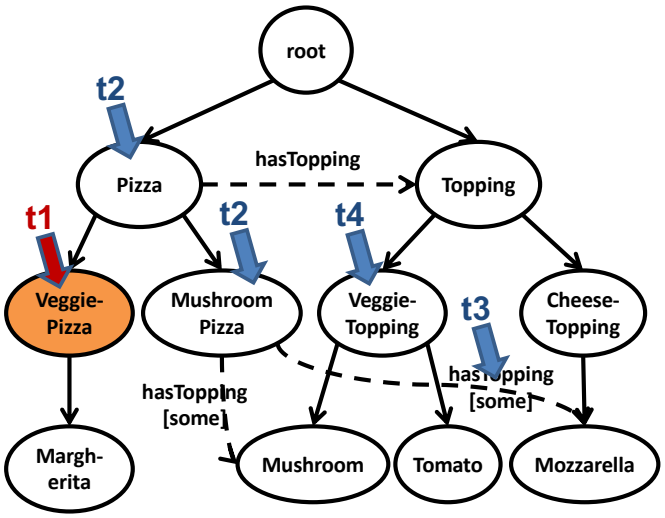


G (original ontology)



G' (changed ontology) with 1 change

Example

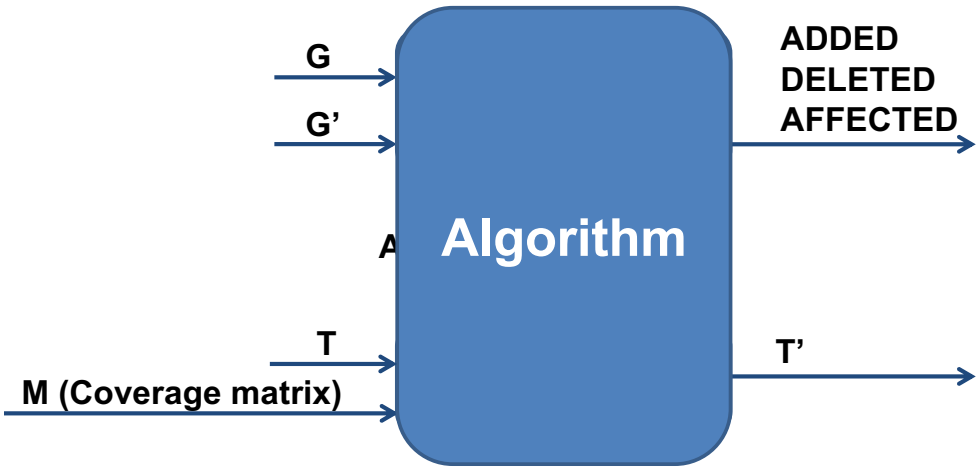


G (original ontology)

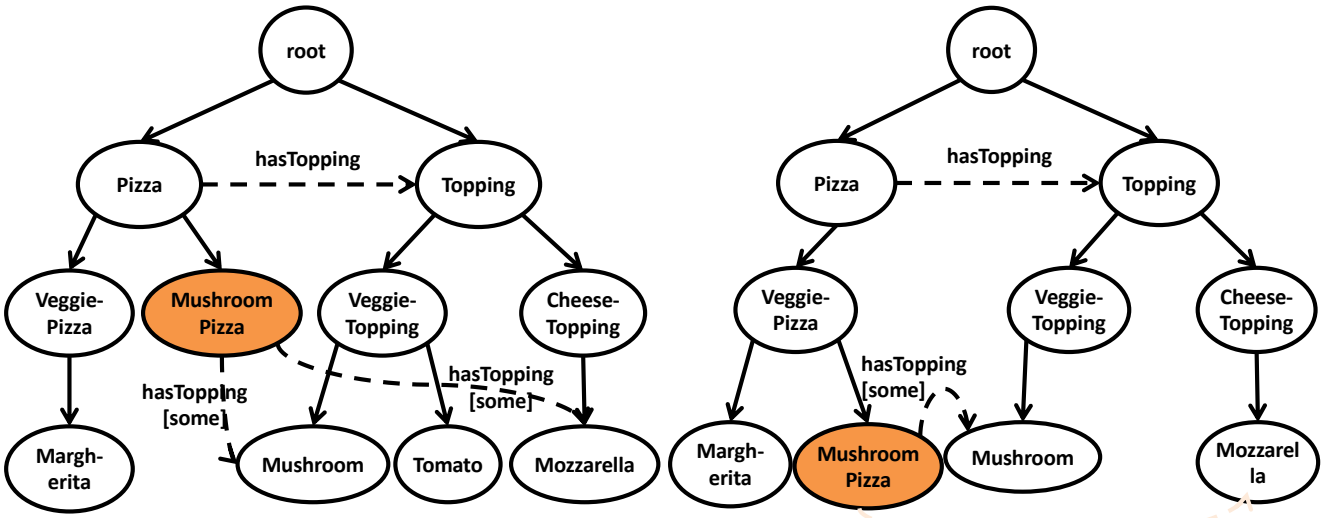
Test Case		Entities
t1	There are at least 3 veggie pizzas	<VeggiePizza>
t2	Pizza must include mushroom pizza	<Pizza>, <MushroomPizza>
t3	Mushroom pizza must have mozzarella toppings	<MushroomPizza hasTopping [some] Mozzarella>
t4	There must be at least 1 veggie topping	<VeggieTopping>

Coverage Matrix

Algorithm



Changes Made to G'

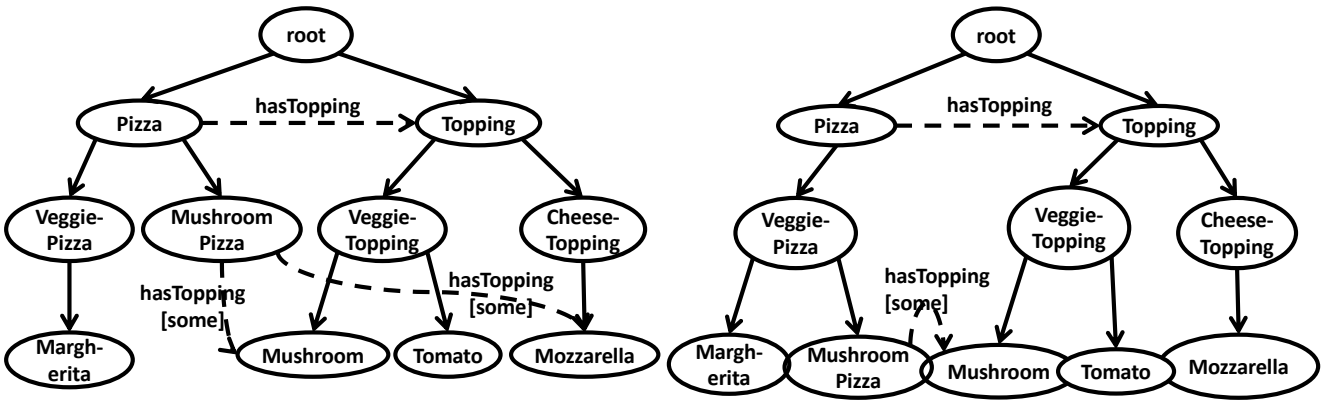


G (original ontology)

G' (changed ontology) with 2 changes

Phase 1: Compute Effects of Changes

for each n∈G
  if match n'∈G' found
    if Descendants(n) ≠ Descendants(n')
      n is AFFECTED by changes
    if OutgoingPropertyEdges(n) ≠ OutgoingPropertyEdges(n')
      property edges(n) DELETED or property edges(n') ADDED
  else n is DELETED //match not found



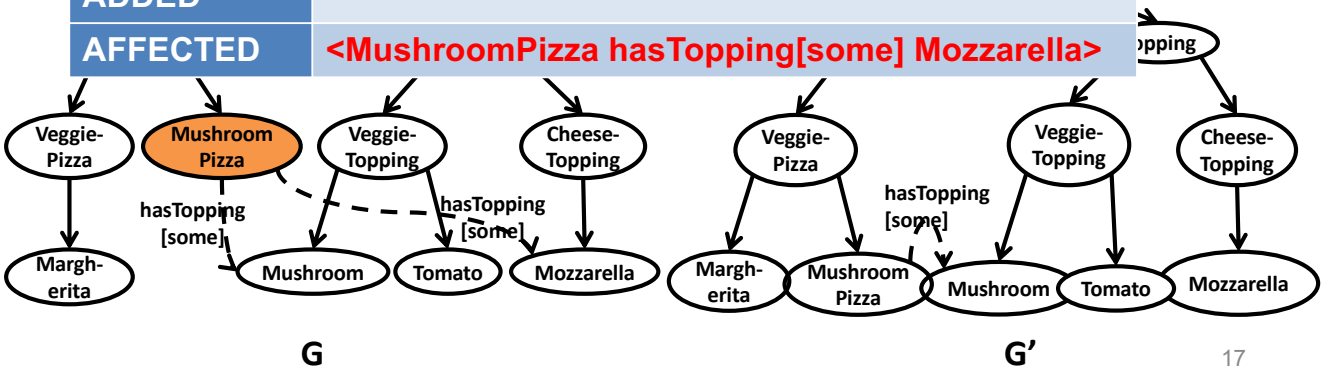
G **G'**

Phase 1: Compute Effects of Changes

when n = MushroomPizza
  if match n'∈G' found
    if Descendants(n) ≠ Descendants(n')

    if OutgoingPropertyEdges(n) ≠ OutgoingPropertyEdges(n')
      add difference to DELETED and AFFECTED

DELETED	**<MushroomPizza hasTopping[some] Mozzarella>**
ADDED	
AFFECTED	**<MushroomPizza hasTopping[some] Mozzarella>**

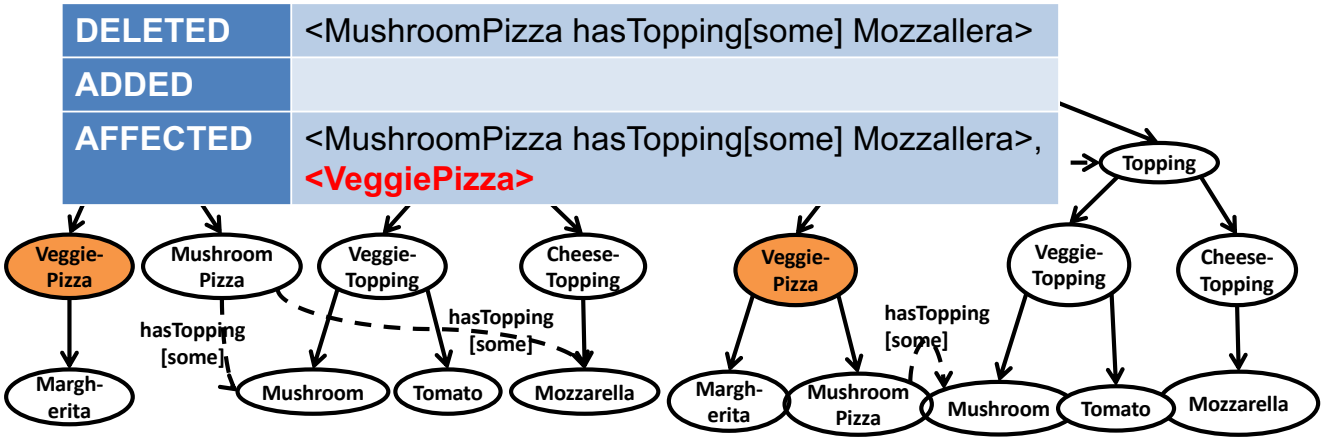


G G'

Phase 1: Compute Effects of Changes

when n = VeggiePizza
  if match n'∈G' found
    if Descendants(n) ≠ Descendants(n')
      add {VeggiePizza} to AFFECTED
    if OutgoingPropertyEdges(n) ≠ OutgoingPropertyEdges(n')

DELETED	<MushroomPizza hasTopping[some] Mozzallera>
ADDED	
AFFECTED	<MushroomPizza hasTopping[some] Mozzallera>, **<VeggiePizza>**

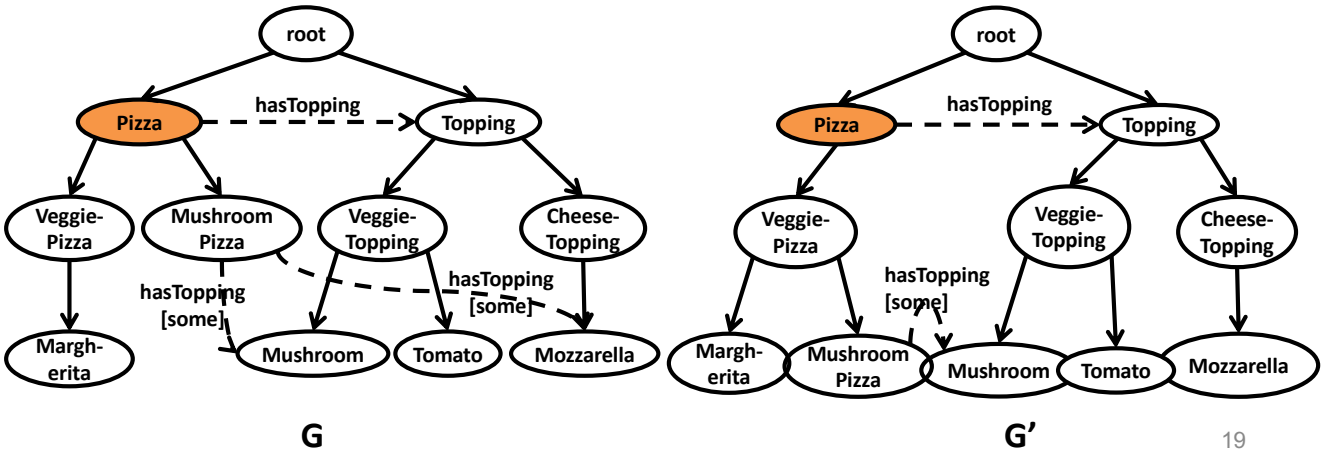


G G'

Phase 1: Compute Effects of Changes

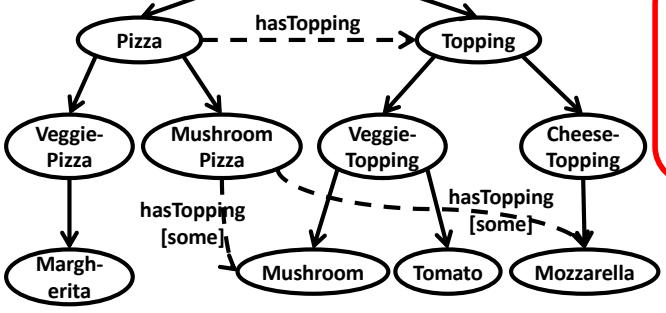
when n = Pizza
  if match n'∈G' found
    if Descendants(n) ≠ Descendants(n')

if OutgoingPropertyEdges(n) ≠ OutgoingPropertyEdges(n')



Phase 2: Select test cases

DELETED	<MushroomPizza hasTopping [some] Mozzarella>
ADDED	
AFFECTED	<MushroomPizza hasTopping [some] Mozzarella>, <VeggiePizza>



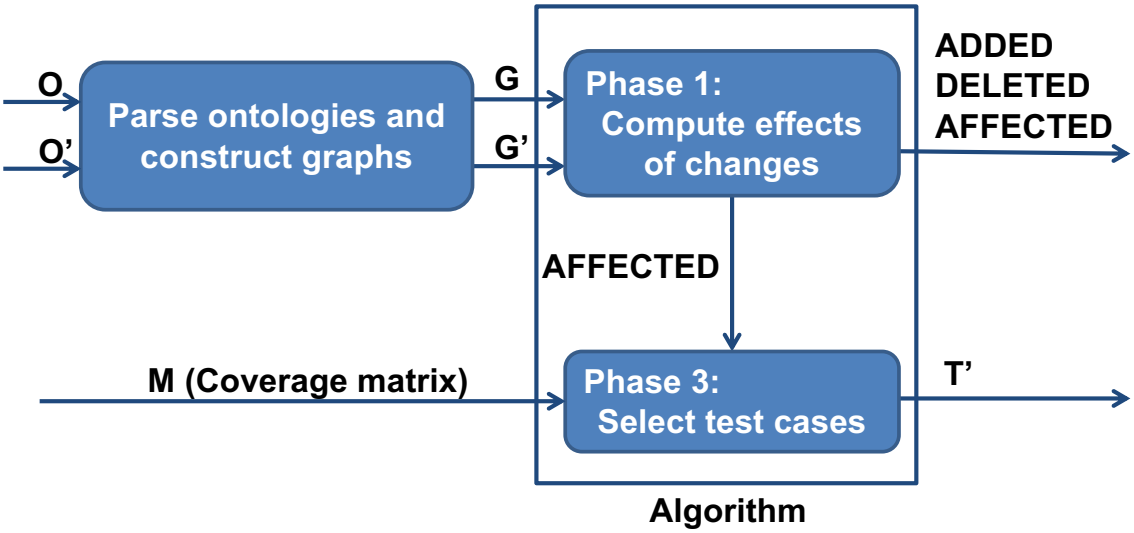
G

T' = {t1, t3}

	Test Case	Entities
t1	There are at least 3 veggie pizzas	<VeggiePizza>
t2	Pizza must include mushroom pizza	<Pizza>, <MushroomPizza>
t3	Mushroom pizza must have mozzarella toppings	<MushroomPizza hasTopping [some] Mozzarella>
t4	There must be at least 1 veggie topping	<VeggieTopping>

Coverage Matrix

Our Tool: OntoReTest



Subjects

- i2b2 (Informatics for Integrating Biology and the Bedside)
  - Large informatics framework
  - Targets data warehouse in clinical domain
- Gene Ontology (GO)
  - Genomic information provided by GO consortium
  - Relational database consisting of ontologies and annotated data with ontologies

Subjects

Subject	Classes	Properties and Restrictions	Versions	Test Cases	Coverage
i2b2	104967	0	7	1331	99.99%
GO	34636	11101	7	3499	52.75%

- 7 versions for both subjects include
  - For i2b2, real changes and 5 intermediate versions
  - For GO, versions provided by public BioPortal
- Test cases used in studies are
  - For i2b2, user queries run on public i2b2 instance
  - For GO, randomly generated queries based on 7 query templates

Study 1: Effectiveness

Goal: To evaluate the effectiveness of our approach in selecting test cases for changes in ontologies

Method

- Ran OntoReTest on each pair of versions
  - Recorded size of AFFECTED
  - Recorded size of T'
- Computed percentage of test cases selected

Results of Study 1

	Version Pair	AFFECTED	Size(T') / Retest all	% of test cases selected
i2b2	(v_0, v_a)	15945	626 / 1331	47.03
	(v_0, v_b)	1553	150 / 1331	11.27
	(v_0, v_c)	9615	136 / 1331	10.22
	(v_0, v_d)	4412	78 / 1331	5.86
	(v_0, v_e)	64	43 / 1331	3.23
	(v_0, v_f)	31581	906 / 1331	68.07
GO	(v_0, v_a)	57	5/ 3499	0.14
	(v_0, v_b)	259	27 / 3499	0.77
	(v_0, v_c)	523	56 / 3499	1.60
	(v_0, v_d)	587	59 / 3499	1.69
	(v_0, v_e)	835	74 / 3499	2.11
	(v_0, v_f)	7214	624 / 3499	17.83

Study 2: Efficiency

Goal: To evaluate the savings in regression time provided by our approach

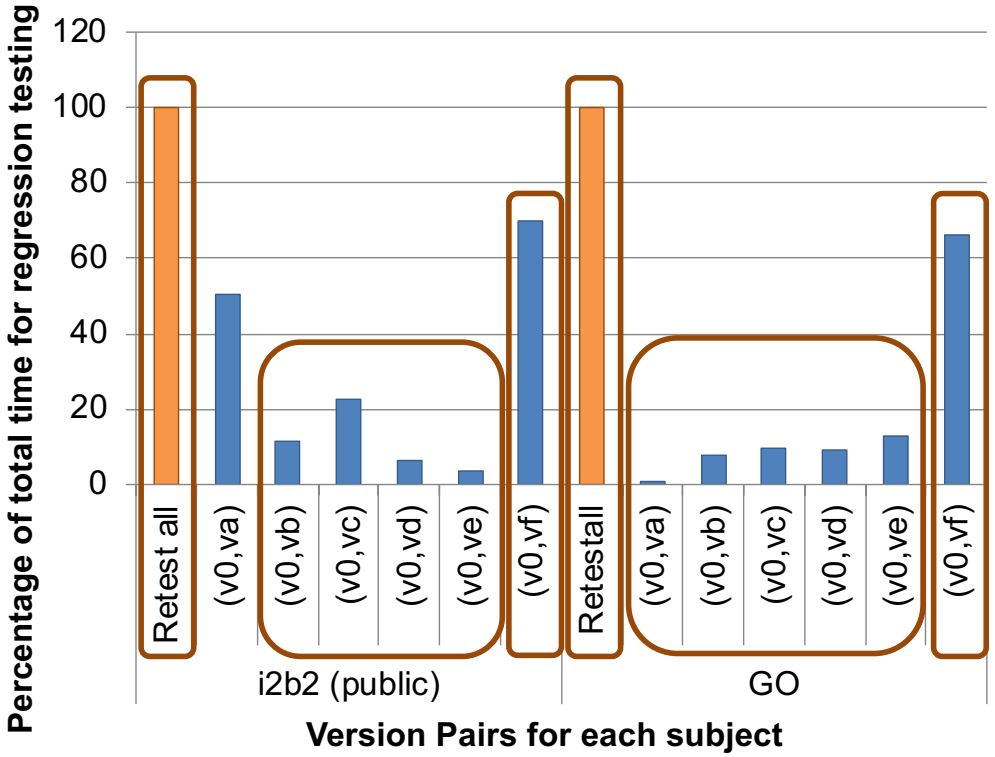
Method

- Computed time for regression testing using Retest-all
- Computed time for regression testing using our approach (Time (OntoReTest) + Time (test cases selected))
- Compared time for regression testing using Retest-all with time for regression testing using our approach

Results of Study 2

Subjects	Version Pair	OntoReTest (secs)	Running time of selected tests (secs)	Total running time (secs)
i2b2	Retest all	n/a	n/a	7528
	(v_0, v_a)	43	3803	3846
	(v_0, v_b)	41	834	875
	(v_0, v_c)	45	822	867
	(v_0, v_d)	40	441	481
	(v_0, v_e)	42	255	297
	(v_0, v_f)	40	5231	5271
GO	Retest all	n/a	n/a	14220
	(v_0, v_a)	96	0.012	96.012
	(v_0, v_b)	89	948	1037
	(v_0, v_c)	101	1014	1115
	(v_0, v_d)	91	1015	1106
	(v_0, v_e)	87	1385	1472
	(v_0, v_f)	95	9451	9546

Results of Study 2



Future Work

- Perform additional experiments
  - More subjects in biomedical domain
  - Subjects from other domains
- Extend technique to support more testing activities
  - Test-case prioritization
  - Test-suite augmentation
  - Obsolete test identification
  - Test-case repair
- Investigate detecting inconsistencies in ontology-driven systems with multiple databases

Contributions

- First RTS technique for ontology-driven systems
  - Creates graph representations
  - Computes affected ontology entities
  - Selects test cases
- OntoReTest, run on two real-world subjects
  - Reduces number of tests to be rerun
  - Reduces time required for regression testing

Questions?