January 22th, 2014

LoRea Group

LSI-EHU Members

- Javier Álvez
- Jose Gaintzarain
- Montse Hermo
- Paqui Lucio
- Marisa Navarro

Ikerbasque

- Hubie Chen

Students

- Javier Albors
- Urko Lopez de Abechuco
- Sergio Rivas

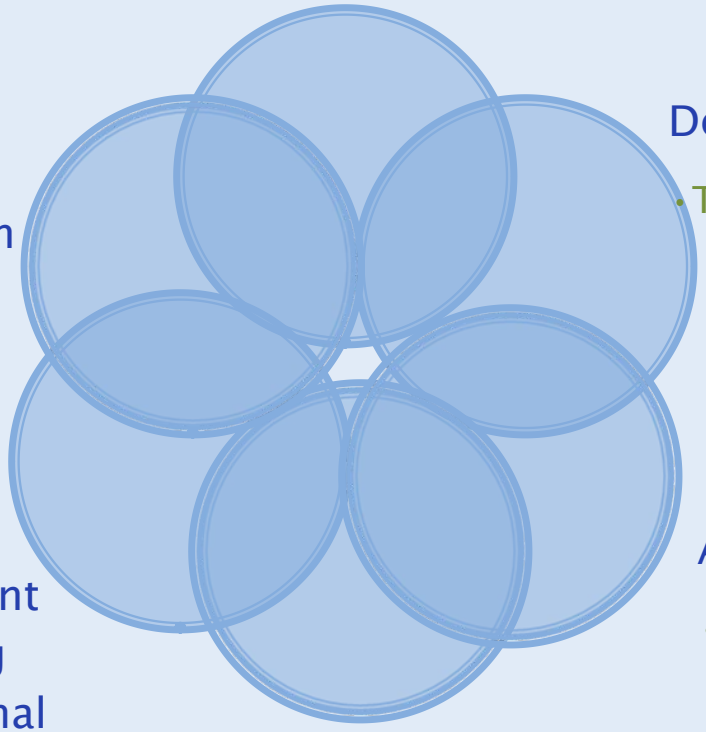
LoRea Current Research Projects

- Métodos formales y algoritmos para el diseño de sistemas (FORMALISM)
  CICYT (REF. TIN2007-66523)

- Logic, Reasoning and Complexity (LoReaCo)
  Gobierno Vasco (ref.-UPV: SAI12/219)

- Logic and Reasoning (LoRea)
  UPV-EHU (ref UPV-EHU GIU12/26)

LoRea Current Research Lines



Temporal Logic

Declarative Programming

- Temporal Logic Programming Languages

Graph Transformation

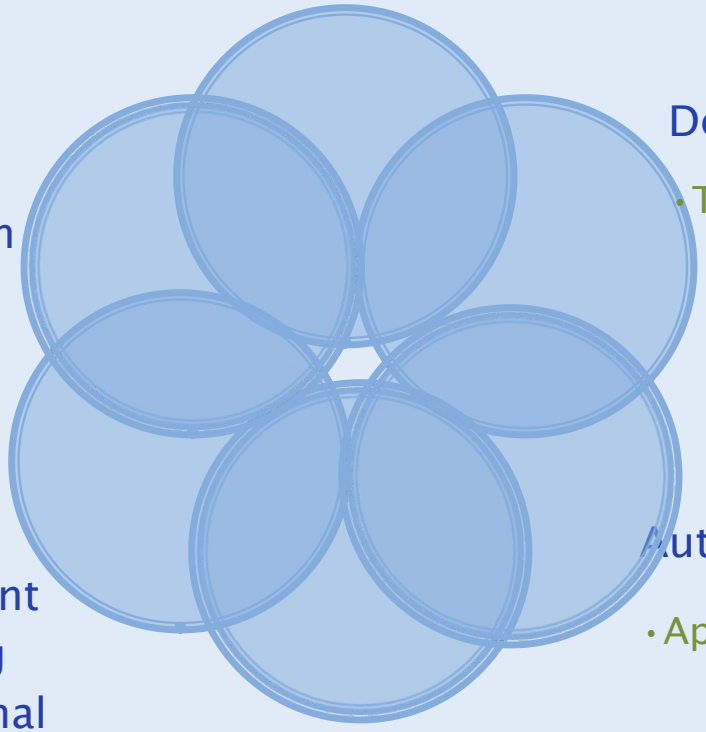
Automated Reasoning

- Applications to Ontological Reasoning

Constraint Solving (Equational Logic)

Computational Complexity

LoRea Current Research Lines



Temporal Logic

Declarative Programming

- Temporal Logic Programming Languages

Graph Transformation

Automated Reasoning

- Applications to Ontological Reasoning

Constraint Solving (Equational Logic)

Computational Complexity

Ontologies

- An ontology is a description of the concepts and relationships that exists in a domain.
- Formal ontologies are (large) theories in some logic
- Ontologies are being used in wide range of applications and knowledge based systems.

Ontological Reasoning

- Almost every type of intelligent task (natural language processing, planning, learning, high-level vision, expert-level reasoning) requires some degree of reasoning to carry out.

- In order to enable better reasoning support to intelligent task, the inference engine should be able to extract us much implicit knowledge as possible.

SUMO Ontology

- Suggested Upper Merged Ontology (SUMO)
  - Browse/English word: Brain/ SUMO Mappings: Brain
- SUMO and its domain ontologies form the largest formal public ontology in existence today.
  - about 25,000 terms and 80,000 axioms
- They are being used for research and applications in search, linguistics and reasoning.
- SUMO is the only formal ontology that has been mapped to all of the WordNet lexicon.

Could an inference engine infer from SUMO the truth or falsity of the assert:

"*plants do not have brain*" ?

```
( =>
   (and
     (instance ?BRAIN Brain)
     (instance ?PLANT Plant))
   (not
     (part ?BRAIN ?PLANT)))
```

$$\forall B\ \forall P\ (\ (instance(B, brain) \wedge instance(P, plant)) \rightarrow\ \neg part(B, P)\ )$$

Adimen-SUMO

- Adimen-SUMO is the off-the-shelf first-order ontology that were presented in

  - Javier Alvez, Paqui Lucio, and German Rigau.

    Adimen-SUMO: Reengineering an ontology for first-order reasoning.

    International Journal on Semantic Web and Information Systems, 8(4):80-116, 2012.

- Adimen-SUMO can be found in

  - http://adimen.si.ehu.es/web/AdimenSUMO.

Adimen-SUMO

- Adimen-SUMO has been obtained by reengineering the 88% of the the large and complex ontology SUMO

- The reengineering work were performed by using first-order theorem provers (concretely Vampire and E-Prover) as inference engines.

Adimen-SUMO

- Now, we automatically infer from Adimen-SUMO that

"plants do not have brain"

```
( =>
    (and
      (instance ?BRAIN Brain)
      (instance ?PLANT Plant))
    (not
      (part ?BRAIN ?PLANT)))
```

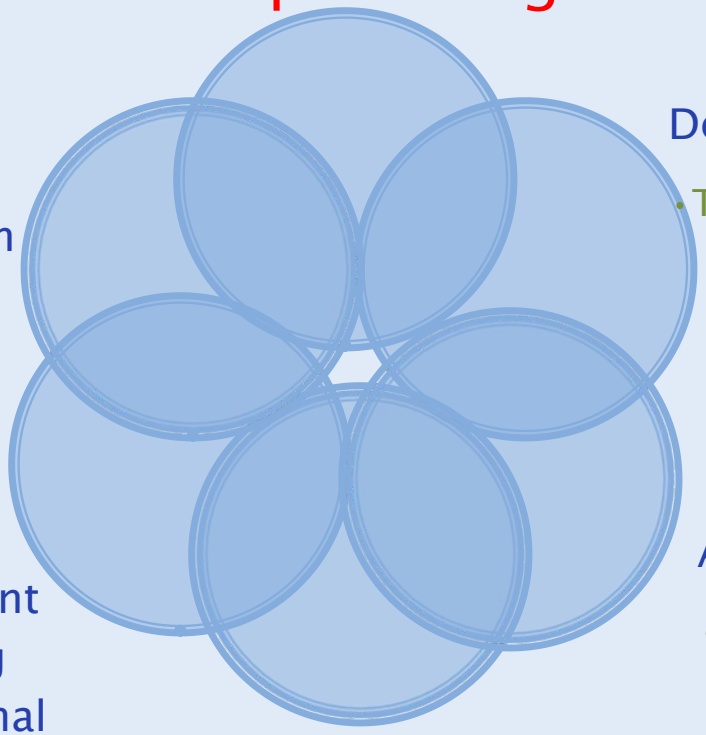
$$\forall B \, \forall P \, (\, (instance(B, brain) \wedge instance(P, plant)) \rightarrow \neg part(B, P) \,)$$

Work in Progress

- Like any other dependable component of a system, an ontology has to go through a repetitive process of refinement and evaluation during its development lifecycle.
  - Benchmark for improvement and validation.

- Current theorem provers and finite model generators are not able to answer: "*Yes, Adimen-Sumo is consistent*"
  - Ad-hoc consistency proof.

LoRea Current Research Lines



Propositional Linear Temporal Logic (PLTL)

Atomic Propositions: p, q, r, …

Boolean Connectives: $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$

Temporal operators

p	"p is true now"
○p	"p is true in the next state"
◊p	"p will be eventually true in the future"
□p	"p will be always true in the future"
q $\mathcal{U}$ p	"q will hold true until p eventually becomes true"

Propositional Linear Temporal Logic (PLTL)

Atomic Propositions: p, q, r, …

Boolean Connectives: $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$

Temporal operators

p	"p is true now"
○p	"p is true in the next state"
◊p	"p will be eventually true in the future"
□p	"p will be always true in the future"
q $\mathcal{U}$ p	"q will hold true until p eventually becomes true"

Propositional Linear Temporal Logic (PLTL)

Atomic Propositions: p, q, r, …

Boolean Connectives: $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$

Temporal operators

p	"p is true now"
○p	"p is true in the next state"
◊p	"p will be eventually true in the future"
□p	"p will be always true in the future"
q $\mathcal{U}$ p	"q will hold true until p eventually becomes true"

Propositional Linear Temporal Logic (PLTL)

Atomic Propositions: p, q, r, …

Boolean Connectives: $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$

Temporal operators

p	"p is true now"
○p	"p is true in the next state"
◊p	"p will be eventually true in the future"
□p	"p will be always true in the future"
q $\mathcal{U}$ p	"q will hold true until p eventually becomes true"

Propositional Linear Temporal Logic (PLTL)

Atomic Propositions: p, q, r, …

Boolean Connectives: $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$

Temporal operators

p	"p is true now"
○p	"p is true in the next state"
◊p	"p will be eventually true in the future"
□p	"p will be always true in the future"
q $\mathcal{U}$ p	"q will hold true until p eventually becomes true"

PLTL Examples

$\Box(\text{warning} \rightarrow \bigcirc \text{backup})$

When a warning occurs, a backup is done at the next instant.

$\Box \Diamond \text{backup}$

Always, a backup will be eventually done.

$\Box(\text{insert-card} \rightarrow \Diamond(\text{open} \vee \text{error}))$

After the user inserts the card, eventually the door is opened or error is displayed.

Automated Reasoning for PLTL

- Inference systems:
  - J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, and F. Orejas

    Dual Systems of Tableaux and Sequents for PLTL

    The Journal of Logic and Algebraic Programming 78: 701–722, (2009).
  - J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, and F. Orejas

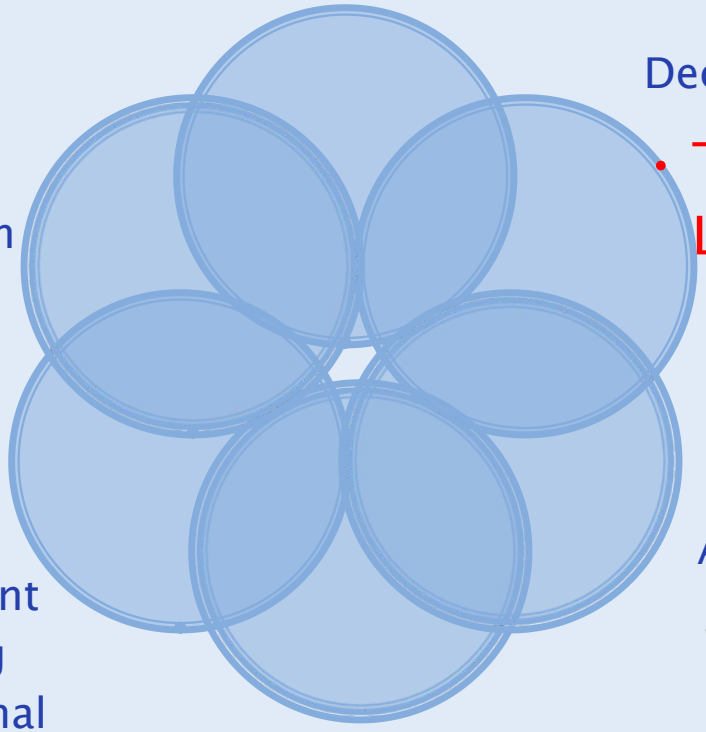
    Invariant-Free Clausal Temporal Resolution

    Journal of Automated Reasoning, Volume 50, Issue 1, 2013.
- Prototypes:
  - TTM: A Tableau-based Theorem Prover for PLTL
  - TRS: A Resolution-based Theorem Prover for PLTL

Work in Progress

- Improving prototypes by combining tableaux and resolution.
- Practical Application of PLTL Theorem Proving
- A cut-free sequent calculus for CTL* would be also a great innovation in TL.
  - for CTL*, there is only an intricate tableau method [Reynolds, 2009] and a complete Hilbert-style system [Reynolds, 2000].

LoRea Current Research Lines



Temporal Logic

Declarative Programming

- Temporal Logic Programming Languages

Graph Transformation

Automated Reasoning

- Applications to Ontological Reasoning

Constraint Solving (Equational Logic)

Computational Complexity

TeDiLog

- A Temporal Disjunctive Logic Programming Language
  - J. Gaintzarain and P. Lucio

    Logical Foundations for More Expressive Declarative Temporal Logic Programming Languages

    ACM Transactions on Computational Logic, *14 (4) ,1–28, 2013.*

- Purely declarative sub-language of PLTL
- Specialized Invariant-Free Resolution for PLTL
- More expressive than every declarative language in the literature (e.g. Templog, Chronolog, Gabbay's Temporal Prolog)

An Example

$\Box\Diamond$ack_sm

$\Box$(ack_sm $\rightarrow\Diamond$eop_dv)

$\Box$(($\Diamond$eop_dv) $\rightarrow$ com_dv)

$\Diamond\neg$com_dv (goal: $\Box$com_dv)

sm = system manager,
dv= device

ack_sm: *the sm sends an acknowledgment giving permission*

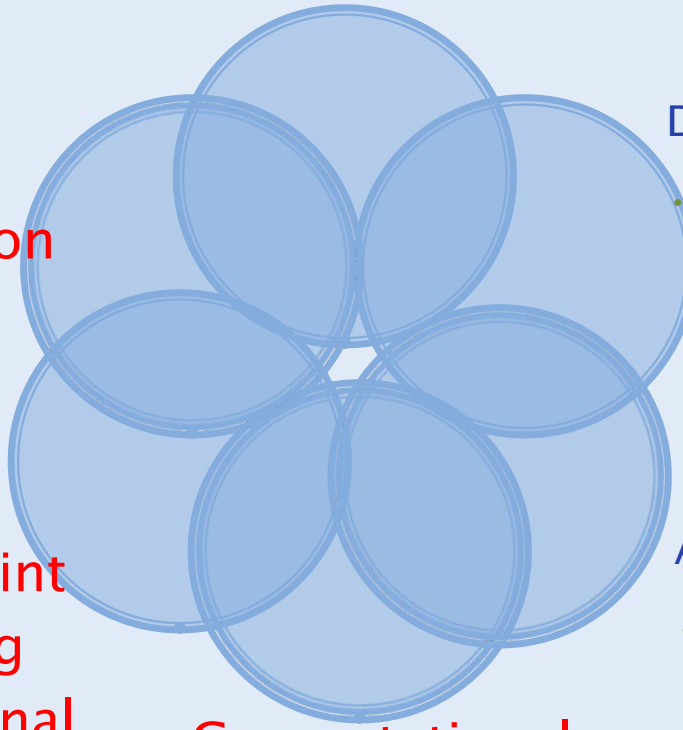
com_dv: *the dv is in communicating state*

eop_dv: *the dv sends a signal of end_of_process*

Future Work

- Implementation of TeDiLog
  - experimentation
  - improvements
  - practical application
- First-order extension of TeDiLog
  - FTL is incomplete, but ...

LoRea Current Research Lines



Temporal Logic

Declarative Programming

- Temporal Logic Programming Languages

Automated Reasoning

- Applications to Ontological Reasoning

Computational Complexity

Constraint Solving (Equational Logic)

Graph Transformation

Thanks !