Description Logic and Ontology

Scott Farrar
CLMA, University of Washington
farrar@u.washington.edu

May 19, 2009

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Background on Description Logics and OWL-DL

RDF

Once upon a time there were a bunch of librarians who needed a way to describe metadata: Topic Maps (ISO) vs. RDF (W3C). **RDF won**. The enterprise continues as the Semantic Web, or most recently, LinkedData...

RDF+DL=OWL

There was a need for a more expressive language. The result: OWL the product of the unhappy marriage between RDF supporters and Description Logic researchers.

OWL w/o RDF

We now see a movement to get away from some of the problems presented by RDF's conceptual model.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Background on Description Logics and OWL-DL

RDF

Once upon a time there were a bunch of librarians who needed a way to describe metadata: Topic Maps (ISO) vs. RDF (W3C). **RDF won**. The enterprise continues as the Semantic Web, or most recently, LinkedData...

RDF+DL=OWL

There was a need for a more expressive language. The result: OWL the product of the unhappy marriage between RDF supporters and Description Logic researchers.

OWL w/o RDF

We now see a movement to get away from some of the problems presented by RDF's conceptual model.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Background on Description Logics and OWL-DL

RDF

Once upon a time there were a bunch of librarians who needed a way to describe metadata: Topic Maps (ISO) vs. RDF (W3C). **RDF won**. The enterprise continues as the Semantic Web, or most recently, LinkedData...

RDF+DL=OWL

There was a need for a more expressive language. The result: OWL the product of the unhappy marriage between RDF supporters and Description Logic researchers.

OWL w/o RDF

We now see a movement to get away from some of the problems presented by RDF's conceptual model.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Description Logics

Description Logic is the formal basis of OWL.

Description Logic (DL) is becoming more and more popular, in no small part, due to the success (or hype) of OWL, but also because of the development of highly optimized reasoning algorithms and tools that implement those algorithms.

That presents a few questions for computational linguists:

- Can we use Description Logic to the benefit of NLP?
- If so, then which DL?
- If so, then for what sub-tasks of NLP?
- How do ontologies fit in with all this?

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Description Logics

Description Logic is the formal basis of OWL.

Description Logic (DL) is becoming more and more popular, in no small part, due to the success (or hype) of OWL, but also because of the development of highly optimized reasoning algorithms and tools that implement those algorithms.

That presents a few questions for computational linguists:

- Can we use Description Logic to the benefit of NLP?
- If so, then which DL?
- If so, then for what sub-tasks of NLP?
- How do ontologies fit in with all this?

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Description Logics

Description Logic is the formal basis of OWL.

Description Logic (DL) is becoming more and more popular, in no small part, due to the success (or hype) of OWL, but also because of the development of highly optimized reasoning algorithms and tools that implement those algorithms.

That presents a few questions for computational linguists:

- Can we use Description Logic to the benefit of NLP?
- If so, then which DL?
- If so, then for what sub-tasks of NLP?
- How do ontologies fit in with all this?

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Description Logics

Description Logic is the formal basis of OWL.

Description Logic (DL) is becoming more and more popular, in no small part, due to the success (or hype) of OWL, but also because of the development of highly optimized reasoning algorithms and tools that implement those algorithms.

That presents a few questions for computational linguists:

- Can we use Description Logic to the benefit of NLP?
- If so, then which DL?
- If so, then for what sub-tasks of NLP?
- How do ontologies fit in with all this?

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Description Logics

Description Logic is the formal basis of OWL.

Description Logic (DL) is becoming more and more popular, in no small part, due to the success (or hype) of OWL, but also because of the development of highly optimized reasoning algorithms and tools that implement those algorithms.

That presents a few questions for computational linguists:

- Can we use Description Logic to the benefit of NLP?
- If so, then which DL?
- If so, then for what sub-tasks of NLP?
- How do ontologies fit in with all this?

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Description Logics

Description Logic is the formal basis of OWL.

Description Logic (DL) is becoming more and more popular, in no small part, due to the success (or hype) of OWL, but also because of the development of highly optimized reasoning algorithms and tools that implement those algorithms.

That presents a few questions for computational linguists:

- Can we use Description Logic to the benefit of NLP?
- If so, then which DL?
- If so, then for what sub-tasks of NLP?
- How do ontologies fit in with all this?

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Description Logics

Description Logic is the formal basis of OWL.

Description Logic (DL) is becoming more and more popular, in no small part, due to the success (or hype) of OWL, but also because of the development of highly optimized reasoning algorithms and tools that implement those algorithms.

That presents a few questions for computational linguists:

- Can we use Description Logic to the benefit of NLP?
- If so, then which DL?
- If so, then for what sub-tasks of NLP?
- How do ontologies fit in with all this?

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

What's the answer?

Sí se puede:

- temporal Description Logic: allow reasoning about time in narrative interpretation
- concept reasoning: allow better named entity recognition/classification
- qualified restrictions: formal way to map lexemes ($Cousin \sqcap Female \equiv Prima$)
- roles: Davidsonian event semantics

Major drawbacks:

- lack of full quantification (*pace* donkey beaters)
- no defined sem. composition operations (e.g., λ-calculus)

Today's talk

1. Intro to DL
   - Varieties of DL
   - DL Knowledge Bases
2. Previous work: DL and NLP
3. Applying DLs to NLP
4. More info

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

25 minute intro to Description Logic

Definition

A **description logic** allows for the *description* of concepts and for a description of relations holding among concept instances. They are furthermore:

- decidable fragments of FOL and have a set theoretic semantics
- closely related to modal logics, e.g., **S5**:
  - $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
- **designed for optimization**, but at the expense of expressibility
- highly theoretical but made practical with a number of solid implementations

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

25 minute intro to Description Logic

Definition

A **description logic** allows for the *description* of concepts and for a description of relations holding among concept instances. They are furthermore:

- decidable fragments of FOL and have a set theoretic semantics
- closely related to modal logics, e.g., **S5**:
  - $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
- **designed for optimization**, but at the expense of expressibility
- highly theoretical but made practical with a number of solid implementations

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

25 minute intro to Description Logic

Definition

A **description logic** allows for the *description* of concepts and for a description of relations holding among concept instances. They are furthermore:

- decidable fragments of FOL and have a set theoretic semantics
- closely related to modal logics, e.g., **S5**:
  - $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
- **designed for optimization**, but at the expense of expressibility
- highly theoretical but made practical with a number of solid implementations

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

25 minute intro to Description Logic

Definition

A **description logic** allows for the *description* of concepts and for a description of relations holding among concept instances. They are furthermore:

- decidable fragments of FOL and have a set theoretic semantics
- closely related to modal logics, e.g., **S5**:
  - $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
- **designed for optimization**, but at the expense of expressibility
- highly theoretical but made practical with a number of solid implementations

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

25 minute intro to Description Logic

Definition

A **description logic** allows for the *description* of concepts and for a description of relations holding among concept instances. They are furthermore:

- decidable fragments of FOL and have a set theoretic semantics
- closely related to modal logics, e.g., **S5**:
  - $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
- **designed for optimization**, but at the expense of expressibility
- highly theoretical but made practical with a number of solid implementations

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

25 minute intro to Description Logic

Definition

A **description logic** allows for the *description* of concepts and for a description of relations holding among concept instances. They are furthermore:

- decidable fragments of FOL and have a set theoretic semantics
- closely related to modal logics, e.g., **S5**:
  - $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
- **designed for optimization**, but at the expense of expressibility
- highly theoretical but made practical with a number of solid implementations

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Terminology note

Concept

The term **concept** is used for classes of individuals. Concepts are related to one another by the relations of subsumption or equivalence.

Individual

The term **individual** refers to an instance of some concept. Individuals are related to concepts via the instantiation relation.

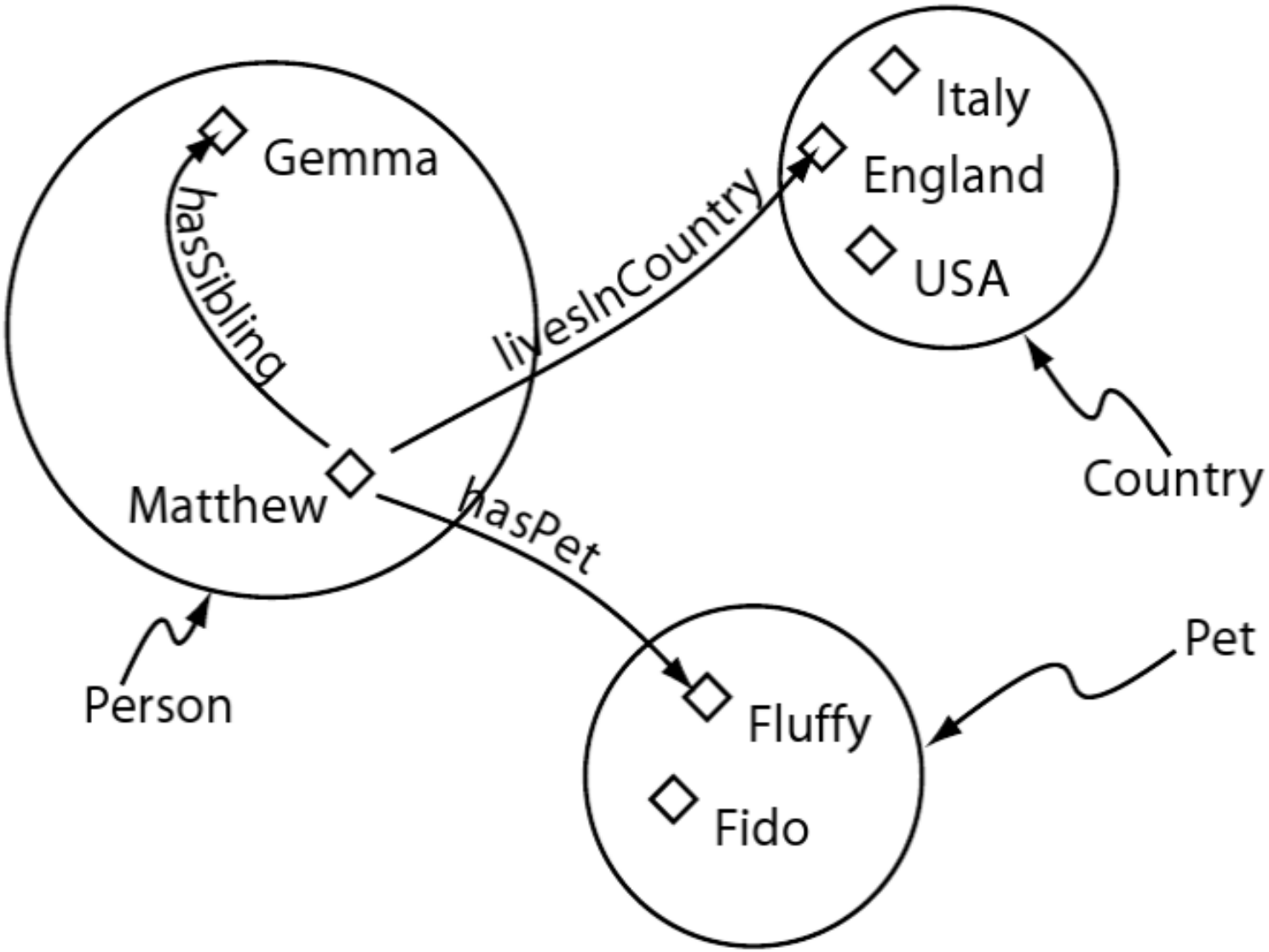
Role

The term **role** refers to a binary relation holding among individuals.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Classes, individuals, and properties



Terminology note

terminology!

DL lit. uses **concept**, OWL lit. uses **class**

DL lit. uses **role**, OWL lit. uses **property**

Named concepts

Example

Doctor, the concept representing doctors
Vehicle, the concept representing vehicles
MotionEvent, the concept representing motion events

Concepts can simply be declared to exist, with or without any information about what belongs to the concept. A **primitive concept** is not defined.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

What's it mean?

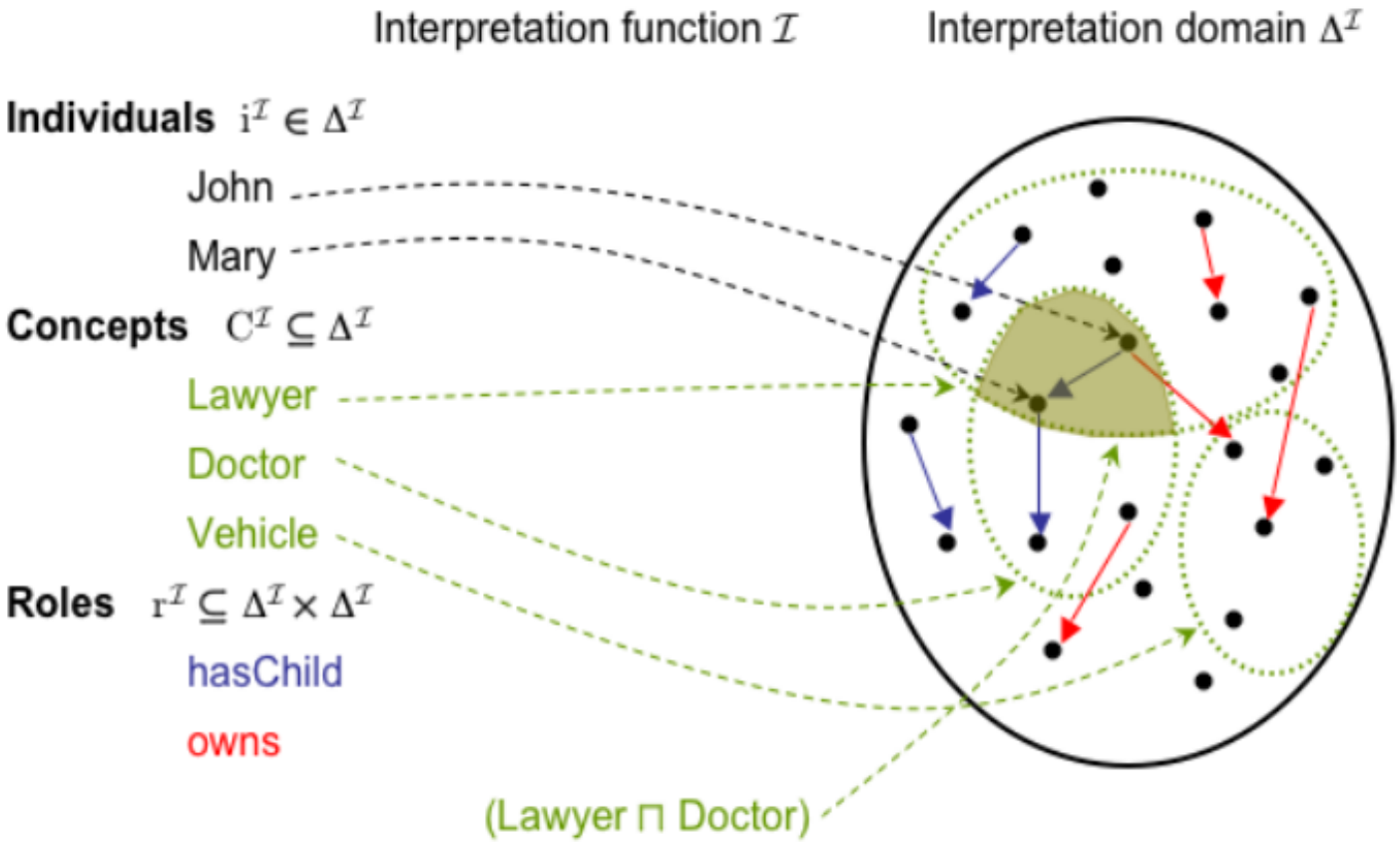
Semantics

- an interpretation $I = (\Delta^I, \cdot^I)$
- nonempty set Δ^I (the domain)
- a function $\cdot^I$ (the interpretation function)
- that maps:
  - every concept to a subset of Δ^I
  - every role to a subset of $\Delta^I \times \Delta^I$

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

DL semantics



Intersection

Definition

An **intersection concept** is described by combining two or more concepts using the $\sqcap$ operator.

Example

Consider the intersection of **Doctor** and **Lawyer**, an anonymous concept that contains the individuals that are members of the concept **Doctor** and the concept **Lawyer**.

Meaning

The semantics of an intersection concept: .

$(Lawyer \sqcap Doctor)^I = Lawyer^I \cap Doctor^I$

Intersection

Definition

An **intersection concept** is described by combining two or more concepts using the $\sqcap$ operator.

Example

Consider the intersection of **Doctor** and **Lawyer**, an anonymous concept that contains the individuals that are members of the concept **Doctor** and the concept **Lawyer**.

Meaning

The semantics of an intersection concept: .

$(Lawyer \sqcap Doctor)^I = Lawyer^I \cap Doctor^I$

Intersection

Definition

An **intersection concept** is described by combining two or more concepts using the $\sqcap$ operator.

Example

Consider the intersection of **Doctor** and **Lawyer**, an anonymous concept that contains the individuals that are members of the concept **Doctor** and the concept **Lawyer**.

Meaning

The semantics of an intersection concept: .
$(Lawyer \sqcap Doctor)^I = Lawyer^I \cap Doctor^I$

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Union

Definition

A **union concept** is created by combining two or more concepts using the $\sqcup$ operator. The process merges individuals from two or more concepts into one concept.

Example

Consider the union of **Man** and **Woman**, describing an anonymous concept that contains the individuals that belong to either the concept **Man** or the concept **Woman** (or both).

Meaning

The semantics of a union concept:
$(Man \sqcup Woman)^I = Man^I \cup Woman^I$

Union

Definition

A **union concept** is created by combining two or more concepts using the $\sqcup$ operator. The process merges individuals from two or more concepts into one concept.

Example

Consider the union of **Man** and **Woman**, describing an anonymous concept that contains the individuals that belong to either the concept **Man** or the concept **Woman** (or both).

Meaning

The semantics of a union concept:

$(Man \sqcup Woman)^I = Man^I \cup Woman^I$

Union

Definition

A **union concept** is created by combining two or more concepts using the $\sqcup$ operator. The process merges individuals from two or more concepts into one concept.

Example

Consider the union of **Man** and **Woman**, describing an anonymous concept that contains the individuals that belong to either the concept **Man** or the concept **Woman** (or both).

Meaning

The semantics of a union concept:
$(Man \sqcup Woman)^I = Man^I \cup Woman^I$

Negation

Definition

A negated concept is created by using the $\neg$ operator. The process excludes individuals that belong to a given concept.

Example

$\neg Professor$, everything that is not a professor
$\neg(Professor \sqcup Miner)$, everything that is not a professor or miner.

Meaning

The semantics of a concept negation is set complement:
$(\neg Professor)^I = \Delta^I \setminus Professor^I$

Beyond $\sqcap, \sqcup, \neg$

Beyond simple set theoretic operations, we can define more complex ways of making **restrictions** on what individuals can be included in a concept

Restrictions

- **Quantifier**: a restriction using either the existential or the universal quantifier
- **Cardinality**: a restriction using cardinality constraints (next time)
- **hasValue**: a restriction that refers to a specific (next time) individual

Beyond $\sqcap, \sqcup, \neg$

Beyond simple set theoretic operations, we can define more complex ways of making **restrictions** on what individuals can be included in a concept

Restrictions

- **Quantifier**: a restriction using either the existential or the universal quantifier
- **Cardinality**: a restriction using cardinality constraints (next time)
- **hasValue**: a restriction that refers to a specific (next time) individual

Beyond $\sqcap, \sqcup, \neg$

Beyond simple set theoretic operations, we can define more complex ways of making **restrictions** on what individuals can be included in a concept

Restrictions

- **Quantifier**: a restriction using either the existential or the universal quantifier
- **Cardinality**: a restriction using cardinality constraints (next time)
- **hasValue**: a restriction that refers to a specific (next time) individual

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Beyond $\sqcap, \sqcup, \neg$

Beyond simple set theoretic operations, we can define more complex ways of making **restrictions** on what individuals can be included in a concept

Restrictions

- **Quantifier**: a restriction using either the existential or the universal quantifier
- **Cardinality**: a restriction using cardinality constraints (next time)
- **hasValue**: a restriction that refers to a specific (next time) individual

Beyond $\sqcap, \sqcup, \neg$

Beyond simple set theoretic operations, we can define more complex ways of making **restrictions** on what individuals can be included in a concept

Restrictions

- **Quantifier**: a restriction using either the existential or the universal quantifier
- **Cardinality**: a restriction using cardinality constraints (next time)
- **hasValue**: a restriction that refers to a specific (next time) individual

Quantifier restrictions

In a DL we can create the concept for individuals that must participate in "at least one" role of a certain type. In other words, there can be no individual without a certain role to a second specified concept.

Examples

- something that has parts that are legos.
  $\exists\ hasPart.Lego$
- something that has a leader that is a lieutenant.
  $\exists\ leader.Lieutenant$
- Those who attend both TreeHouse meetings and colloquiums
  $Person \sqcap \exists attends.(THMeeting \sqcup Colloq)$

Quantifier restrictions

In a DL we can create the concept for individuals that must participate in "at least one" role of a certain type. In other words, there can be no individual without a certain role to a second specified concept.

Examples

- something that has parts that are legos.
  $\exists\ hasPart.Lego$
- something that has a leader that is a lieutenant.
  $\exists\ leader.Lieutenant$
- Those who attend both TreeHouse meetings and colloquiums
  $Person \sqcap \exists attends.(THMeeting \sqcup Colloq)$

Description Logic and Ontology

Scott Farrar CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Quantifier restrictions

In a DL we can create the concept for individuals that must participate in "at least one" role of a certain type. In other words, there can be no individual without a certain role to a second specified concept.

Examples

- something that has parts that are legos.
  $\exists\ hasPart.Lego$
- something that has a leader that is a lieutenant.
  $\exists\ leader.Lieutenant$
- Those who attend both TreeHouse meetings and colloquiums
  $Person \sqcap \exists attends.(THMeeting \sqcup Colloq)$

Description Logic and Ontology

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Quantifier restrictions

In a DL we can create the concept for individuals that must participate in "at least one" role of a certain type. In other words, there can be no individual without a certain role to a second specified concept.

Examples

- something that has parts that are legos.
  $\exists\ hasPart.Lego$
- something that has a leader that is a lieutenant.
  $\exists\ leader.Lieutenant$
- Those who attend both TreeHouse meetings and colloquiums
  $Person \sqcap \exists attends.(THMeeting \sqcup Colloq)$

Quantifier restrictions

In a DL we can create the concept for individuals that must participate in "at least one" role of a certain type. In other words, there can be no individual without a certain role to a second specified concept.

Examples

- something that has parts that are legos.
  $\exists\ hasPart.Lego$
- something that has a leader that is a lieutenant.
  $\exists\ leader.Lieutenant$
- Those who attend both TreeHouse meetings and colloquiums
  $Person \sqcap \exists attends.(THMeeting \sqcup Colloq)$

Quantifier restrictions

something that has gills
$\exists$ *hasOrgan.Gill*

Semantics

$(\exists hasOrgan.Gill)^I = \{x \in \Delta^I \mid \exists y.(x, y) \in hasOrgan^I \wedge y \in Gill^I\}$

Quantifier restrictions

something that has gills
$\exists$ *hasOrgan.Gill*

Semantics

$(\exists hasOrgan.Gill)^I = \{x \in \Delta^I \mid \exists y.(x, y) \in hasOrgan^I \wedge y \in Gill^I\}$

Quantifier restrictions

We can also create the concept of individuals that participate in only given types of relationships. It's not really quantification, but a *value restrictor*.

Examples

- those that have regenerable limbs
  $\forall\ hasLimb.Regenerable$
- those that have CLMA students as students
  $\forall\ hasStudent.CLMAStudent$
- those that only possess thrift store items.
  $\forall\ possess.ThriftStoreItem$

Semantics

$$(\forall possess.ThriftStoreItem)^I = \{x \in \Delta^I \mid \forall y.(x,y) \in possess^I \Rightarrow y \in ThriftStoreItem^I\}$$

Quantifier restrictions

We can also create the concept of individuals that participate in only given types of relationships. It's not really quantification, but a *value restrictor*.

Examples

- those that have regenerable limbs
  $\forall$ *hasLimb*.*Regenerable*
- those that have CLMA students as students
  $\forall$ *hasStudent*.*CLMAStudent*
- those that only possess thrift store items.
  $\forall$ *possess*.*ThriftStoreItem*

Semantics

$(\forall possess.ThriftStoreItem)^I = \{x \in \Delta^I \mid \forall y.(x,y) \in possess^I \Rightarrow y \in ThriftStoreItem^I\}$

Quantifier restrictions

We can also create the concept of individuals that participate in only given types of relationships. It's not really quantification, but a *value restrictor*.

Examples

- those that have regenerable limbs
  $\forall$ *hasLimb*.*Regenerable*
- those that have CLMA students as students
  $\forall$ *hasStudent*.*CLMAStudent*
- those that only possess thrift store items.
  $\forall$ *possess*.*ThriftStoreItem*

Semantics

$(\forall possess.ThriftStoreItem)^I = \{x \in \Delta^I \mid \forall y.(x,y) \in possess^I \Rightarrow y \in ThriftStoreItem^I\}$

Quantifier restrictions

We can also create the concept of individuals that participate in only given types of relationships. It's not really quantification, but a *value restrictor*.

Examples

- those that have regenerable limbs
  $\forall$ *hasLimb*.*Regenerable*
- those that have CLMA students as students
  $\forall$ *hasStudent*.*CLMAStudent*
- those that only possess thrift store items.
  $\forall$ *possess*.*ThriftStoreItem*

Semantics

$(\forall possess.ThriftStoreItem)^I = \{x \in \Delta^I \mid \forall y.(x,y) \in possess^I \Rightarrow y \in ThriftStoreItem^I\}$

Quantifier restrictions

We can also create the concept of individuals that participate in only given types of relationships. It's not really quantification, but a *value restrictor*.

Examples

- those that have regenerable limbs
  $\forall$ *hasLimb*.*Regenerable*
- those that have CLMA students as students
  $\forall$ *hasStudent*.*CLMAStudent*
- those that only possess thrift store items.
  $\forall$ *possess*.*ThriftStoreItem*

Semantics

$$(\forall possess.ThriftStoreItem)^I = \{x \in \Delta^I \mid \forall y.(x,y) \in possess^I \Rightarrow y \in ThriftStoreItem^I\}$$

Quantifier restrictions

We can also create the concept of individuals that participate in only given types of relationships. It's not really quantification, but a *value restrictor*.

Examples

- those that have regenerable limbs
  $\forall$ *hasLimb*.*Regenerable*
- those that have CLMA students as students
  $\forall$ *hasStudent*.*CLMAStudent*
- those that only possess thrift store items.
  $\forall$ *possess*.*ThriftStoreItem*

Semantics

$(\forall possess.ThriftStoreItem)^I = \{x \in \Delta^I \mid \forall y.(x,y) \in possess^I \Rightarrow y \in ThriftStoreItem^I\}$

Expressivity

DLs can be made less or more expressive by adding particular *features* to the language:

- negation $\neg Student$
- role transitivity Trans(*hasAncestor*)
- concrete data types (e.g., string, integer, date)
- quantification $\exists$, $\forall$
- ...

This is important b/c reasoners can be optimized for certain DLs.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Expressivity

DLs can be made less or more expressive by adding particular *features* to the language:

- negation $\neg Student$
- role transitivity Trans(*hasAncestor*)
- concrete data types (e.g., string, integer, date)
- quantification $\exists$, $\forall$
- ...

This is important b/c reasoners can be optimized for certain DLs.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Expressivity

DLs can be made less or more expressive by adding particular *features* to the language:

- negation $\neg Student$
- role transitivity Trans(*hasAncestor*)
- concrete data types (e.g., string, integer, date)
- quantification $\exists$, $\forall$
- ...

This is important b/c reasoners can be optimized for certain DLs.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Expressivity

DLs can be made less or more expressive by adding particular *features* to the language:

- negation $\neg Student$
- role transitivity Trans(*hasAncestor*)
- concrete data types (e.g., string, integer, date)
- quantification $\exists$, $\forall$
- ...

This is important b/c reasoners can be optimized for certain DLs.

Expressivity

DLs can be made less or more expressive by adding particular *features* to the language:

- negation $\neg Student$
- role transitivity Trans(*hasAncestor*)
- concrete data types (e.g., string, integer, date)
- quantification $\exists$, $\forall$
- ...

This is important b/c reasoners can be optimized for certain DLs.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Expressivity

DLs can be made less or more expressive by adding particular *features* to the language:

- negation $\neg Student$
- role transitivity Trans(*hasAncestor*)
- concrete data types (e.g., string, integer, date)
- quantification $\exists$, $\forall$
- ...

This is important b/c reasoners can be optimized for certain DLs.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Expressivity

DLs can be made less or more expressive by adding particular *features* to the language:

- negation $\neg Student$
- role transitivity Trans(*hasAncestor*)
- concrete data types (e.g., string, integer, date)
- quantification $\exists$, $\forall$
- ...

This is important b/c reasoners can be optimized for certain DLs.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

DL are named using what kinds of features they support:

- $\mathcal{C}$ for (atomic) concept negation
  e.g., $\neg$ Professor
- $\mathcal{H}$ for role hierarchy
  e.g., hasDaughter subrole of hasChild
- $\mathcal{U}$ for Union $\sqcup$
- $\mathcal{O}$ for nominals/singleton concepts
  e.g., { Italy,Germany }
- $\mathcal{I}$ for inverse roles
  e.g., isChildOf, hasChild

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

DL are named using what kinds of features they support:

- $\mathcal{C}$ for (atomic) concept negation
  e.g., $\neg$ Professor
- $\mathcal{H}$ for role hierarchy
  e.g., hasDaughter subrole of hasChild
- $\mathcal{U}$ for Union $\sqcup$
- $\mathcal{O}$ for nominals/singleton concepts
  e.g., { Italy,Germany }
- $\mathcal{I}$ for inverse roles
  e.g., isChildOf, hasChild

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

DL are named using what kinds of features they support:

- $\mathcal{C}$ for (atomic) concept negation
  e.g., $\neg$ Professor
- $\mathcal{H}$ for role hierarchy
  e.g., hasDaughter subrole of hasChild
- $\mathcal{U}$ for Union $\sqcup$
- $\mathcal{O}$ for nominals/singleton concepts
  e.g., { Italy,Germany }
- $\mathcal{I}$ for inverse roles
  e.g., isChildOf, hasChild

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

DL are named using what kinds of features they support:

- $\mathcal{C}$ for (atomic) concept negation
  e.g., $\neg$ Professor
- $\mathcal{H}$ for role hierarchy
  e.g., hasDaughter subrole of hasChild
- $\mathcal{U}$ for Union $\sqcup$
- $\mathcal{O}$ for nominals/singleton concepts
  e.g., { Italy,Germany }
- $\mathcal{I}$ for inverse roles
  e.g., isChildOf, hasChild

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

DL are named using what kinds of features they support:

- $\mathcal{C}$ for (atomic) concept negation
  e.g., $\neg$ Professor
- $\mathcal{H}$ for role hierarchy
  e.g., hasDaughter subrole of hasChild
- $\mathcal{U}$ for Union $\sqcup$
- $\mathcal{O}$ for nominals/singleton concepts
  e.g., { Italy,Germany }
- $\mathcal{I}$ for inverse roles
  e.g., isChildOf, hasChild

Description Logic and Ontology

Scott Farrar CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

DL are named using what kinds of features they support:

- $\mathcal{C}$ for (atomic) concept negation
  e.g., $\neg$ Professor
- $\mathcal{H}$ for role hierarchy
  e.g., hasDaughter subrole of hasChild
- $\mathcal{U}$ for Union $\sqcup$
- $\mathcal{O}$ for nominals/singleton concepts
  e.g., { Italy,Germany }
- $\mathcal{I}$ for inverse roles
  e.g., isChildOf, hasChild

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

- $\mathcal{N}$ for number restrictions
  e.g., > 2 hasChild
- $\mathcal{Q}$ for qualified number restrictions
  e.g., > 2 hasChild.Doctor
- $\mathcal{F}$ for functional roles
- **D** for concrete data types
  string, integer, float

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

- $\mathcal{N}$ for number restrictions
  e.g., > 2 hasChild
- $\mathcal{Q}$ for qualified number restrictions
  e.g., > 2 hasChild.Doctor
- $\mathcal{F}$ for functional roles
- **D** for concrete data types
  string, integer, float

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

- $\mathcal{N}$ for number restrictions
  e.g., > 2 hasChild
- $\mathcal{Q}$ for qualified number restrictions
  e.g., > 2 hasChild.Doctor
- $\mathcal{F}$ for functional roles
- **D** for concrete data types
  string, integer, float

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

- $\mathcal{N}$ for number restrictions
  e.g., > 2 hasChild
- $\mathcal{Q}$ for qualified number restrictions
  e.g., > 2 hasChild.Doctor
- $\mathcal{F}$ for functional roles
- **D** for concrete data types
  string, integer, float

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Some DL naming conventions

- $\mathcal{N}$ for number restrictions
  e.g., > 2 hasChild
- $\mathcal{Q}$ for qualified number restrictions
  e.g., > 2 hasChild.Doctor
- $\mathcal{F}$ for functional roles
- **D** for concrete data types
  string, integer, float

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

DL naming convention

Example

$\mathcal{S}$ + role hierarchy $\mathcal{H}$ + inverse $\mathcal{I}$ + QNR $\mathcal{Q} = \mathcal{SHIQ}$

- $\mathcal{S}$ for $\mathcal{ALC}$ + transitive roles
- $\mathcal{AL}$ for 'attributive language'

DL naming convention

Example

$\mathcal{S}$ + role hierarchy $\mathcal{H}$ + inverse $\mathcal{I}$ + QNR $\mathcal{Q} = \mathcal{SHIQ}$

- $\mathcal{S}$ for $\mathcal{ALC}$ + transitive roles
- $\mathcal{AL}$ for 'attributive language'

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

DL naming convention

Example

$\mathcal{S}$ + role hierarchy $\mathcal{H}$ + inverse $\mathcal{I}$ + QNR $\mathcal{Q} = \mathcal{SHIQ}$

- $\mathcal{S}$ for $\mathcal{ALC}$ + transitive roles
- $\mathcal{AL}$ for 'attributive language'

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

DL naming convention

Example

$\mathcal{S}$ + role hierarchy $\mathcal{H}$ + inverse $\mathcal{I}$ + QNR $\mathcal{Q} = \mathcal{SHIQ}$

- $\mathcal{S}$ for $\mathcal{ALC}$ + transitive roles
- $\mathcal{AL}$ for 'attributive language'

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

The least expressive

$\mathcal{FL}^-$

The smallest Description Logic is called $\mathcal{FL}^-$, "FL minus".

- operator: $\sqcap$
- quantifiers: $\forall\, R.C$, $\exists\, R.\top$

More expressive

$\mathcal{ALC}$

This is another family of DLs, a bit more expressive than $\mathcal{FL}^-$:

- operators: $\sqcap$, $\sqcup$, $\neg$
- quantifiers: $\forall\, R.C$, $\exists\, R.\top$, $\exists\, R.C$
- equivalence: $C \equiv D$,
- subsumption: $C \sqsubseteq D$

Examples of equivalence axioms

$$
\begin{aligned}
\text{Woman} &\equiv \text{Person} \sqcap \text{Female} \\
\text{Man} &\equiv \text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female}) \\
\text{Mother} &\equiv (\text{Person} \sqcap \text{Female}) \sqcap \exists \text{hasChild}.\text{Person} \\
\text{Father} &\equiv (\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \sqcap \exists \text{hasChild}.\text{Person} \\
\text{Parent} &\equiv ((\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \sqcap \exists \text{hasChild}.\text{Person}) \\
&\quad \sqcup ((\text{Person} \sqcap \text{Female}) \sqcap \exists \text{hasChild}.\text{Person}) \\
\text{Grandmother} &\equiv ((\text{Person} \sqcap \text{Female}) \sqcap \exists \text{hasChild}.\text{Person}) \\
&\quad \sqcap \exists \text{hasChild}.(((\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \\
&\qquad \sqcap \exists \text{hasChild}.\text{Person}) \\
&\qquad \sqcup ((\text{Person} \sqcap \text{Female}) \\
&\qquad\quad \sqcap \exists \text{hasChild}.\text{Person})) \\
\text{MotherWithManyChildren} &\equiv ((\text{Person} \sqcap \text{Female}) \sqcap \exists \text{hasChild}.\text{Person}) \sqcap \geqslant 3\ \text{hasChild} \\
\text{MotherWithoutDaughter} &\equiv ((\text{Person} \sqcap \text{Female}) \sqcap \exists \text{hasChild}.\text{Person}) \\
&\quad \sqcap \forall \text{hasChild}.(\neg(\text{Person} \sqcap \text{Female})) \\
\text{Wife} &\equiv (\text{Person} \sqcap \text{Female}) \\
&\quad \sqcap \exists \text{hasHusband}.(\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female}))
\end{aligned}
$$

A DL Knowledge Base

Within a description logic system, **concepts** and **roles** are separated from **individuals** by partitioning the **knowledge base** into two disjoint sets of entities.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

A DL Knowledge Base

Within a description logic system, **concepts** and **roles** are separated from **individuals** by partitioning the **knowledge base** into two disjoint sets of entities.

Definition

A description logic knowledge base KB may be defined minimally as a TBox T and an ABox A, i.e. $KB = \{T, A\}$, where:

A DL Knowledge Base

Within a description logic system, **concepts** and **roles** are separated from **individuals** by partitioning the **knowledge base** into two disjoint sets of entities.

Definition

A description logic knowledge base KB may be defined minimally as a TBox T and an ABox A, i.e. $KB = \{T, A\}$, where:

- T is the union of the set of concepts with the set of roles in the domain

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

A DL Knowledge Base

Within a description logic system, **concepts** and **roles** are separated from **individuals** by partitioning the **knowledge base** into two disjoint sets of entities.

Definition

A description logic knowledge base KB may be defined minimally as a TBox T and an ABox A, i.e. $KB = \{T, A\}$, where:

- T is the union of the set of concepts with the set of roles in the domain
- A is the set of individuals in the domain.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Tbox and Abox

Tbox

Short for *terminology box*, the Tbox contains the concepts and roles and axioms about the domain in general, in the form of logical sentences, e.g., what concepts can be related by what role.

Abox

Short for *assertion box*, the Abox contains the individuals and consists of facts about individuals, mostly in the form of expressions showing particular relations between individuals.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Tbox and Abox

Tbox

Short for *terminology box*, the Tbox contains the concepts and roles and axioms about the domain in general, in the form of logical sentences, e.g., what concepts can be related by what role.

Abox

Short for *assertion box*, the Abox contains the individuals and consists of facts about individuals, mostly in the form of expressions showing particular relations between individuals.

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Why separate?

By separating concepts/roles from individuals, more effective and efficient reasoners can be built, for instance, by focusing only on concepts (Tbox reasoning), or only on individuals (Abox reasoning).

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Today's talk

1. Intro to DL
   - Varieties of DL
   - DL Knowledge Bases
2. Previous work: DL and NLP
3. Applying DLs to NLP
4. More info

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Brief (incomplete) survey of lit.

These papers concern NLP and DL (or OWL):

- Toral et al. (2008): Lexically-based Ontologies and Ontologically Based Lexicons
- Toral and Monachini (2007): SIMPLE-OWL: a Generative Lexicon Ontology for NLP and the Semantic Web
- Wilcock (2007): An OWL ontology for HPSG
- Mellish and Pan (2006): How to name objects in an NLG system
- Franconi (1994): A detailed work on quantifiers, plurals, beliefs
- Fehrer et al. (1994): Earley DL work as part of a dialogue system
- Bateman (1992): Theoretical status of ontologies in NLP (focus on NLG)

Today's talk

1. Intro to DL
   - Varieties of DL
   - DL Knowledge Bases
2. Previous work: DL and NLP
3. Applying DLs to NLP
4. More info

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Today's talk

1. Intro to DL
   - Varieties of DL
   - DL Knowledge Bases
2. Previous work: DL and NLP
3. Applying DLs to NLP
4. More info

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

If you want to know more

- Main DL info page
  http://dl.kr.org/
- Enrico Franconi's tutorial/course pages
  http://www.inf.unibz.it/ franconi/dl/course/
- A list of reasoners (2007)
  http://www.cs.man.ac.uk/ sattler/reasoners.html
- Explore DL complexity/expressiveness applet:
  http://www.cs.man.ac.uk/ ezolin/dl/

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Intro to DL
Varieties of DL
DL Knowledge Bases
Previous work: DL and NLP
Applying DLs to NLP
More info
References

Bateman, J. A. (1992). The theoretical status of ontologies in natural language processing. In Preuß S. and Schmitz, B., editors, *Text Representation and Domain Modelling – ideas from linguistics and AI*, pages 50–99, Berlin, Germany. KIT-Report 97, Technische Universität Berlin. (Papers from KIT-FAST Workshop, Technical University Berlin, October 9th - 11th 1991).

Fehrer, D., Hustadt, U., Jaeger, M., Nonnengart, A., Ohlbach, H. J., Schmidt, R. A., Weidenbach, C., and Weydert, E. (1994). Description logics for natural language processing. In Baader, F., Lenzerini, M., Nutt, W., and Patel-Schneider, P. F., editors, *International Workshop on Description Logics '94*. DFKI.

Franconi, E. (1994). Description logics for natural language processing. In *Proceedings of the 1994 AAAI Fall Symposium on Knowledge Representation for Natural Language Processing in Implemented Systems*, New Orleans, US.

Mellish, C. and Pan, J. (2006). Finding subsumers for natural language presentation. Presented at DL2006 International Workshop on Description Logics, Windermere, England.

Toral, A. and Monachini, M. (2007). SIMPLE-OWL: a generative lexicon ontology for nlp and the semantic web. In *Proceedings of the Workshop of Cooperative Construction of Linguistic Knowledge Bases, 10th Congress of Italian Association for Artificial Intelligence. Rome (Italy)*.

Toral, A., Quochi, V., Gratta, R. D., Monachini, M., Soria, C., and Calzolari, N. (2008). Lexically-based ontologies and ontologically based lexicons. In *Proceedings of the 10th Congress of Italian Association for Artificial Intelligence. Cagliari (Italy)*, pages 49–59.

Wilcock, G. (2007). An OWL ontology for HPSG. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Companion Volume, ACL-2007, Prague*, pages 169–172. ACL.