Representation of Sentence Meaning

Representing Meaning

- **Meaning Representation**: Capturing the meaning of of linguistic utterances using formal notation.
- **Meaning Representation Languages**: Frameworks that are used to specify the syntax and semantics of these meaning representations.
- **Semantic Analysis**: Mapping the linguistic utterances to these meaning representations.

- Correct meaning representation should be selected for the application.
- For certain language tasks require some form of semantic processing:
  - following a recipe
  - answering an essay question in exam
  - ...

Meaning Representation Languages

- Let us look at four frequently used meaning representation languages.
  - First Order Predicate Calculus
  - Semantic Network
  - Conceptual Dependency Diagram
  - Frame-Based Representation

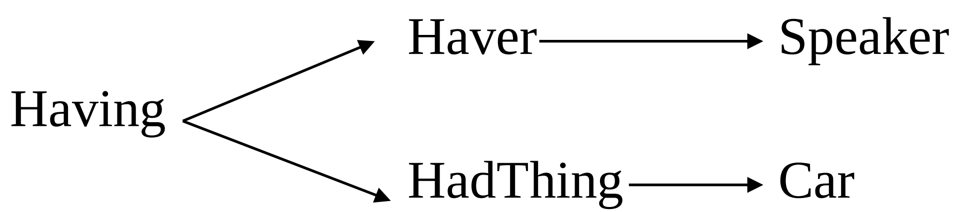
- Let us look at the representation of "*I have a car*" in these four formalism.

Meaning Representation Example - *I have a car*

***First Order Predicate Calculus*:**

$$\exists x,y \, Having(x) \wedge Haver(Speaker,x) \wedge HadThing(y,x) \wedge Car(y)$$

***Semantic Network*:**



***Conceptual Dependency*:**

***Frame-Based Representation*:**

Having
  Haver: Speaker
  HadThing: Car

What do We Expect from Meaning Representations

- To be computationally effective, we expect certain properties in meaning representations:
  - **Verifiability --** Ability to determine the truth value of the representation.
  - **Unambiguous Representations --** A representation must be unambiguous.
  - **Canonical Form --** Utterances which means the same thing should map to the same meaning representation.
  - **Inference and Variables --** Ability to draw valid conclusions based on the meaning representations of inputs and the background knowledge.
  - **Expressiveness --** Ability to express wide range of subject matter.

Verifiability

- **Verifiability --** Ability to determine the truth value of the representation by looking at the information available in the knowledge base.

- **Example:**
  - Assume that we have the entry *serve(Subway,VegetarianFood)* in our KB.
  - Question: *Does Subway serve vegetarian food*?
  - The question should be converted into a logical form (a meaning representation).
  - We should able to verify the truth value of the logical form of the question against our KB.

Unambiguous Representations

- **Unambiguous Representations** -- A meaning representation must be unambiguous.

- **Example**:
  - Assume that we are looking the representation of "*I want to eat some place near Bilkent*".
  - There will be different meanings of this sentence, and we will prefer one of them.
  - But that chosen meaning representation CANNOT be ambiguous.

- **Vagueness**: Vagueness can make it difficult to determine meaning representation, but it does not cause multiple representations.
  - *I want to eat Turkish food*.
  - Here *Turkish food* is vague, but it does not cause multiple representations.
  - Meaning representations should be able to maintain a certain level of vagueness.

Canonical Form

- Distinct inputs can map to the same meaning representation.
  - Does Kirac have vegetarian food?
  - Do they have vegetarian food at Kirac?
  - Are vegetarian dishes served at Kirac?
- We shouldn’t map these sentences to different meaning representations.
- **Canonical Form --** The notion that inputs that mean same thing should have the same meaning representation.
- To able to map distinct inputs to the same meaning representation, we should able to know that different phrases mean the same thing such as *vegetarian food* and *vegetarian dishes*.

Inference and Variables

- **Inference --** Ability to draw valid conclusions based on the meaning representations of inputs and the background knowledge.
- We should be able to find the truth value of propositions that are not explicitly in KB -- *inference*.
- Example:
  - *I would like to find a restaurant that serves vegetarian food*.
  - This example is complex and we should use **variables** in its representation.
  - *serves(x,VegetarianFood)* **--** a part of our meaning representation
  - If there is a restaurant serves vegetarian food, our inference mechanism should be able to find it by binding the variable x to that restaurant.

Expressiveness

- **Expressiveness --** Ability to express wide range of subject matter.
- The ideal situation: a single meaning representation language that could adequately represent the meaning of any sensible natural language utterance.
- Although this ideal situation may not be possible, but the **first order predicate calculus (FOPC)** is expressive enough to handle a lot of things.
- In fact, it is claimed that anything can be representable with other three representation language, it can be also representable with FOPC.
- We will concentrate on FOPC, but other representation languages are also used.
  - For example, Text Meaning Representation (TMR) used in the machine translation system of NMSU is a frame based representation.

Predicate-Argument Structure

- All natural languages have a form of **predicate-argument** arrangement at the core of their semantic structure.
- Specific relations hold among the constituent words and phrases of the sentence. (predicate and its arguments)
- Our meaning representation should support the predicate-argument structure induced by the language.
- In fact, there is a relation between syntactic frames and semantic frames. We will try to find these relations between syntactic frames and semantic frames.
- Example:
  - *Want(somebody,something)* -- Want is predicate with two arguments

Predicate-Argument Structure (cont.)

- Syntactic Structures:
  - *I want Turkish food.* NP want NP
  - *I want to spend less than five dollars.* NP want InfVP
  - *I want it to be close by here.* NP want NP InfVP
- Verb sub-categorization rules allow the linking of the arguments of syntactic structures with the **semantic roles** of these arguments in the semantic representation of that sentence.
  - The study of semantic roles associated with verbs is known as **thematic role.**
- In syntactic structures, there are restrictions on the categories of their arguments.
- Similarly, there are also **semantic restrictions** on the arguments of the predicates.
- The **selectional restrictions** specify semantic restrictions on the arguments of verbs.

Predicate-Argument Structure (cont.)

- Other objects (other than verbs) in natural languages may have predicate-argument structure.

  A Turkish restaurant under fifteen dollars.

  Under(TurkishRestaurant,$15)

  - meaning representation is associated with the preposition *under*.
  - The preposition under can be characterized by a two-argument predicate.

  Make a reservation for this evening for a table for two persons at 8.

  Reservation(Hearer,Today,8PM,2)

  - meaning representation is associated with the noun reservation (not with make).

- Our meaning representation should support :
  - variable arity predicate-argument structures
  - the semantic labeling of arguments to predicates
  - semantic constraints on the fillers of argument roles.

First Order Predicate Calculus (FOPC)

- **First Order Predicate Calculus (FOPC)** is a *flexible*, *well-understood*, and *computationally tractable* approach.
- So, FOPC satisfies the most of the things that we expect from a meaning representation language.
- FOPC provides a sound computational basis for *verifiability*, *inference*, and *expressiveness* requirements.
- The most attractive feature of FOPC is that it makes very few specific commitments for how things should be represented.

Structure of FOPC

$$Formula \rightarrow AtomicFormula \mid Formula\ Connective\ Formula \mid$$
$$Quantifier\ Variable,\ldots\ Formula \mid \neg Formula \mid (Formula)$$

$$AtomicFormula \rightarrow Predicate(Term,\ldots)$$

$$Term \rightarrow Function(Term,\ldots) \mid Constant \mid Variable$$

$$Connective \rightarrow \wedge \mid \vee \mid \Rightarrow$$

$$Quantifier \rightarrow \forall \mid \exists$$

$$Constant \rightarrow \mathrm{A} \mid \mathrm{VegetarianFood} \mid \mathrm{TurkishRestuarant} \mid \ldots$$

$$Variable \rightarrow \mathrm{x} \mid \mathrm{y} \mid \ldots$$

$$Predicate \rightarrow \mathrm{Serves} \mid \mathrm{Want} \mid \mathrm{Under} \mid \ldots$$

$$Function \rightarrow \mathrm{LocationOf} \mid \mathrm{CuisineOf} \mid ...$$

FOPC Example

I only have five dollars and I don’t have a lot of time.

$$Have(Speaker,FiveDollars) \wedge \neg Have(Speaker,LotOfTime)$$

A restaurant that serves Turkish food near Bilkent.

$$\exists x\ Restaurant(x) \wedge Serves(x,TurkishFood) \wedge$$
$$Near(LocationOf(x),LocationOf(Bilkent))$$

All vegetarian restaurants serve vegetarian food.

$$\forall x\ VegetarianRestuarant(x) \Rightarrow Serves(x,VegetarianFood)$$

Semantics of FOPC

- The truth value of each FOPC formula can be computed using meanings of the elements of FOPC.
  - Truth tables for $\neg$ $\wedge$ $\vee$ $\Rightarrow$
  - Meanings of $\forall$ $\exists$
  - Assigned meanings to Predicates, Constant, Functions in an interpretation.

- The truth values of our examples:
  - Have(Speaker,FiveDollars) $\wedge$ $\neg$Have(Speaker,LotOfTime)
  - $\exists$x Restaurant(x) $\wedge$ Serves(x,TurkishFood) $\wedge$
    Near(LocationOf(x),LocationOf(Bilkent))
  - $\forall$x VegetarianRestuarant(x) $\Rightarrow$ Serves(x,VegetarianFood)

Inference

- Ability to determine the truth value of a formula not explicitly contained in a KB.
- We should have **inference rules** to infer new formulas from formulas available in a KB.
- For example, **modes ponens** is a inference rule.

$$\frac{\alpha \qquad \alpha \Rightarrow \beta}{\beta}$$

- Example:

$$\frac{VegetarianRestaurant(Kirac) \qquad \forall x\ VegetarianRestuarant(x) \Rightarrow Serves(x,VegetarianFood)}{Serves(Kirac,VegetarianFood)}$$

Inference (cont.)

- We may use **forward chaining** or **backward chaining** in the implementations of inference rules.
- Implementation of certain inference rules for FOPC is not computationally effective.
- **Resolution** is a computationally effective inference rule.
  - **Prolog** uses resolution and backward chaining.
- Inference rules must be **sound** and **complete**.
  - Sound -- If a formula is derivable using inference rules, it must be valid
  - Complete -- If a formula is valid, it must be derivable.

Inference -- Prolog Example

- Prolog uses resolution and backward chaining.

```
father(X,Y) :- parent(X,Y), male(X).
parent(john,bill).
parent(mary,bill).
male(john).
female(mary).

?- father(F,bill).
```

Representation of Categories

- The semantics of the arguments are expressed in the form of selectional restrictions.
- These selectional restrictions are expressed in the form of **semantically-based categories.**
- The most common way to represent a category is to create a unary predicate.
  - VegaterianRestraunt(Kirac)
  - Here categories are relations (not objects), and difficult to make assertions about categories.
  - We cannot use MostPopular(Kirac,VegetarianRestraunt) because VegetarianRestraunt is not an object.
  - The arguments of formulas must be Terms (Predicates cannot be arguments ın FOPC).

Representation of Categories -- Reification

- Solution is to make each category an object.
  - This technique is know as **reification**.
- Thus we can define relations between objects and categories and relations between categories.
- Membership relation ISA between objects and categories.

  ISA(Kirac,VegetarianRestraunt)

- A category inclusion relation AKO between categories.

  AKO(VegetarianRestraunt,Restraunt)

Representations of Events

- The simplest approach to predicate-argument representation of a verb is to have the same number of arguments present in that verb's subcategorization frame.
- But this simple approach may cause some difficulties:
  - determining correct number of arguments.
  - Ensuring soundness and completeness
- Example:

I ate.	Eating1(Speaker)
I ate a turkey sandwich	Eating2(Speaker,TurkeySandwich)
I ate a turkey sandwich at my desk.	Eating3(Speaker,TurkeySandwich,Desk)
I ate at my desk.	Eating4(Speaker,Desk)
I ate lunch.	Eating5(Speaker,Lunch)
I ate a turkey sandwich for lunch.	Eating6(Speaker,TurkeySandwich,Lunch)
I ate a turkey sandwich for lunch at my desk.	Eating7(Speaker,TurkeySandwich,Lunch,Desk)

Representations of Events -- Another Approach

- Using the maximum number of the arguments and the existential quantifiers will not solve the problem.

I ate at my desk.	$\exists x,y$ Eating(Speaker,x,y,Desk)
I ate lunch.	$\exists x,y$ Eating(Speaker,x,Lunch,y)
I ate lunch at my desk.	$\exists x$ Eating(Speaker,x,Lunch,Desk)

- If we know that 1st and 2nd formulas represent the same event, they can be combined as 3rd formula. But we cannot do this, because we cannot relate events in this approach.

Representations of Events -- A Solution

- We employ **reification** to elevate events to objects.

I ate. -- $\exists x\ ISA(x,Eating) \wedge Eater(x,Speaker)$

I ate a turkey sandwich -- $\exists x\ ISA(x,Eating) \wedge Eater(x,Speaker) \wedge Eaten(x,TurkeySandwich)$

I ate at my desk. -- $\exists x\ ISA(x,Eating) \wedge Eater(x,Speaker) \wedge PlaceEaten(x,Desk)$

I ate lunch. -- $\exists x\ ISA(x,Eating) \wedge Eater(x,Speaker) \wedge MealEaten(x,Lunch)$

- With the reified-event approach:
  - There is no need to specify a fixed number of arguments
  - Many roles can be glued when they appear in the input.
  - We do not need to define relations between different versions of eating (**postulate**)

Representations of Time

- Time flows forward, and the events are asocaiated with either points or intervals in time.
- An ordering among events can be gotten by putting them on the timeline.
- There can be different schemas for represesenting this kind of **temoral information**. (the study of *temporal logic*)
- The **tense of a sentence** will correspond to an ordering of events related with that sentence. (the study of *tense logic*)

Representations of Time -- Example

1. *I arrived in Ankara.* 2. *I am arriving in Ankara.* 3. *I will arrive in Ankara.*

- All three sentences can be represented with the following formula without any temporal information.

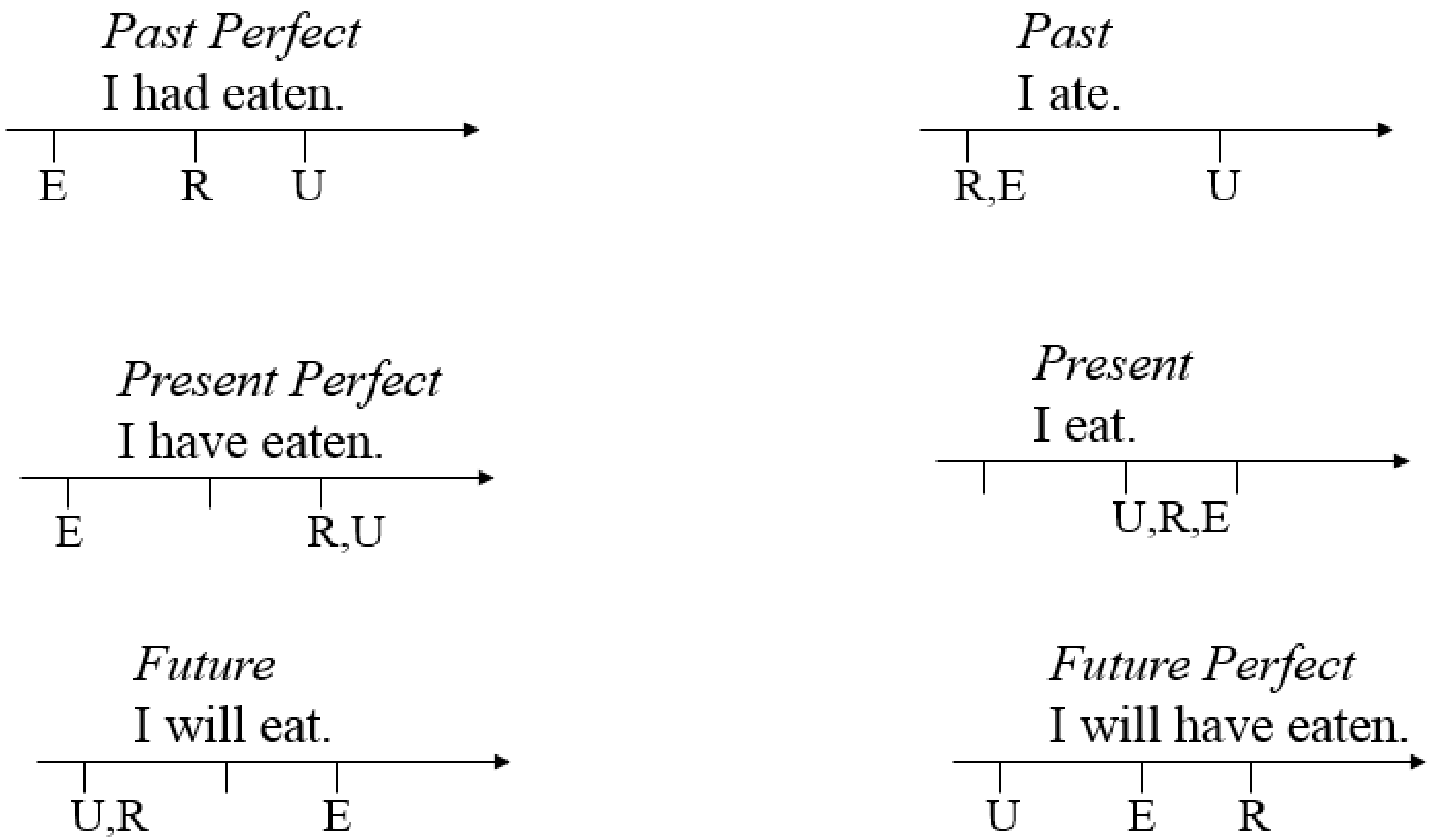
$$\exists w\ ISA(w,Arriving) \wedge Arriver(w,Speaker) \wedge Destination(w,Ankara)$$

- We can add the following representations of temporal information to represent the tenses of these examples.
  1. $\exists w,i,e\ ISA(w,Arriving) \wedge Arriver(w,Speaker) \wedge Destination(w,Ankara) \wedge IntervalOf(w,i) \wedge EndPoint(i,e) \wedge Precedes(e,Now)$
  2. $\exists w,i,e\ ISA(w,Arriving) \wedge Arriver(w,Speaker) \wedge Destination(w,Ankara) \wedge IntervalOf(w,i) \wedge MemberOf(i,Now)$
  3. $\exists w,i,e\ ISA(w,Arriving) \wedge Arriver(w,Speaker) \wedge Destination(w,Ankara) \wedge IntervalOf(w,i) \wedge EndPoint(i,e) \wedge Precedes(Now,e)$

Representations of Time (cont.)

- The relation between simple verb tenses and points in time is not straightforward.
  - We fly from Ankara to Istanbul. **--** present tense refers to a future event
  - Flight 12 will be at gate an hour now. **--** future tense refers to a past event
- In some formalisms, the tense of a sentence is expressed with the relation among ***times of events*** in that sentence, ***time of a reference point***, and ***time of utterance***.

Reinhenbach's Approach to Representing Tenses



Representations of Beliefs

- We can represent a belief as follows:
  - I believe that Mary ate Turkish food.

  $\exists$u,v ISA(u,Believing) $\wedge$ ISA(v,Eating) $\wedge$ Believer(u,Speaker) $\wedge$ Believed(u,v) $\wedge$
  Eater(v,Mary) $\wedge$ Eaten(v,TurkishFood)

- But from this, we can get the following (which may not be correct).

  $\exists$v ISA(v,Eating) $\wedge$ Eater(v,Mary) $\wedge$ Eaten(v,TurkishFood)

- We may think that we can represent this as follows, but it will not be a FOPC formula.

  Believing(Speaker,Eating(Mary,TurkishFood))

- A solution is to augment FOPC with operators. (**modal logic** with **modal operators**).

  Believing(Speaker, $\exists$v ISA(v,Eating) $\wedge$ Eater(v,Mary) $\wedge$ Eaten(v,TurkishFood))

- Inference will be complicated with modal logic.

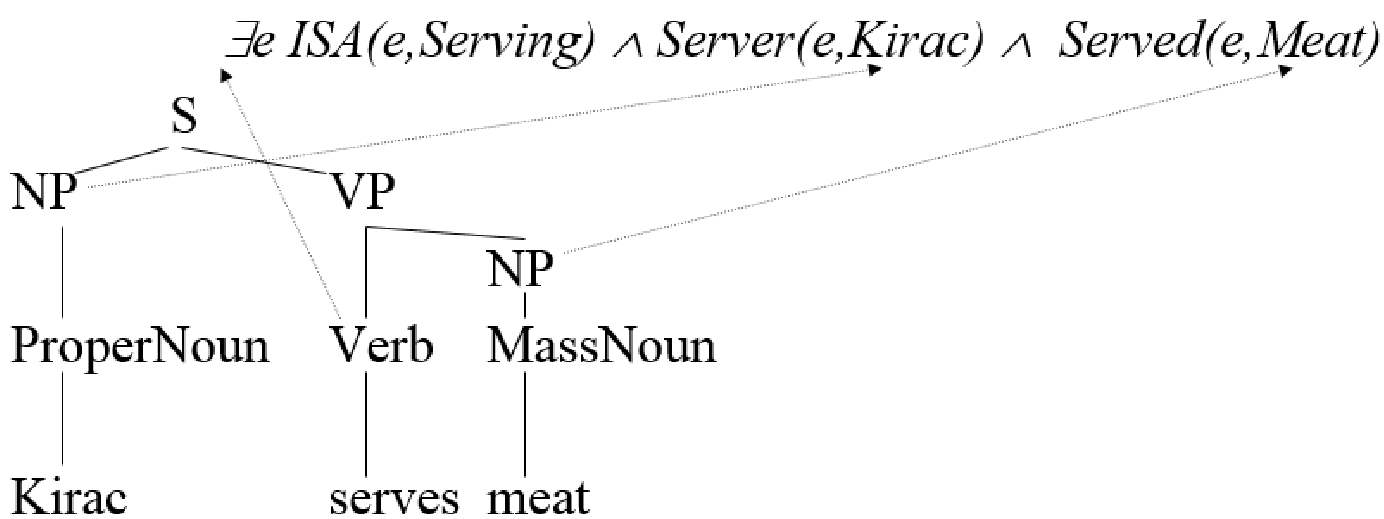
Semantic Analysis

- **Semantic Analysis --** Meaning representations are assigned to linguistic inputs.
- We need static knowledge from grammar and lexicon.
- How much semantic analysis do we need?
  - **Deep Analysis --** Through syntactic and semantic analysis of the text to capture all pertinent information in the text.
  - **Information Extraction --** does not require complete syntactic and semantic analysis. With a cascade of FSAs to produce a robust semantic analyzer.

Syntax-Driven Semantic Analysis

- **Principle of Compositionality --** the meaning of a sentence can be composed of meanings of its parts.
- Ordering and groupings will be important.

Kirac serves meat.



Semantic Augmentation to CFG Rules

- CFG Rules are attached with **semantic attachments**.
- These semantic attachments specify how to compute the meaning representation of a construction from the meanings of its constituent parts.

- A CFG rule with semantic attachment will be as follows:

$$\mathbf{A \rightarrow \alpha_1,\ldots,\alpha_n \qquad \{\ f(\alpha_j.sem,\ldots,\alpha_k.sem)\ \}}$$

- The meaning representation of A, **A.sem**, will be calculated by applying function f to the semantic representations of some constituents.

Naïve Approach

ProperNoun $\rightarrow$ Kirac	{ Kirac }
MassNoun $\rightarrow$ meat	{ Meat }
NP $\rightarrow$ ProperNoun	{ ProperNoun.sem }
NP $\rightarrow$ MassNoun	{ MassNoun.sem }
Verb $\rightarrow$ serves	$\{\exists e,x,y\ ISA(e,Serving) \wedge Server(e,x) \wedge Served(e,y)\}$

- But we cannot propagate this representation to upper levels.

Using Lambda Notations

ProperNoun → Kirac { Kirac }

MassNoun → meat { Meat }

NP → ProperNoun { ProperNoun.sem }

NP → MassNoun { MassNoun.sem }

Verb → serves { $\lambda x \lambda y \, \exists e \; ISA(e,Serving) \land Server(e,y) \land Served(e,x)$ }

VP → Verb NP { Verb.sem(NP.sem) }

S → NP VP { VP.sem(NP.sem) }

application of lambda expression ***lambda expression***

Quasi-Logical Form

- During semantic analysis, we may use quantified expressions as terms. In this case, our formula will not be a FOPC formula.
- We call this form of formulas as **quasi-logical form**.
- A quasi-logical form should be converted into a normal FOPC formula by applying simple syntactic translations.

$Server(e,<\exists x\ ISA(x,Restaurant)>)$ a quasi-logical formula

$$\Downarrow$$

$\exists x\ ISA(x,Restaurant) \wedge Server(e,x)$ a normal FOPC formula

Integrating Semantic Analysis into Earley Algorithm

- Modifications required to integrate a semantic analysis into an Earley parser are:
  - The rules of the grammar will have an extra field to hold semantic attachments.
  - The states in the chart will have an extra field to hold the meaning representation of the constituent.
  - The ENQUEUE function will be changed so that when a complete state is entered into the chart its semantics are computed and stored in the state's semantic field.