# If You Build It, It Will (Hopefully) Parse: A Symbolic Formalism for Embodied Grammar

Nathan Schneider CS 182/CogSci 110/Ling 109 Spring 2008

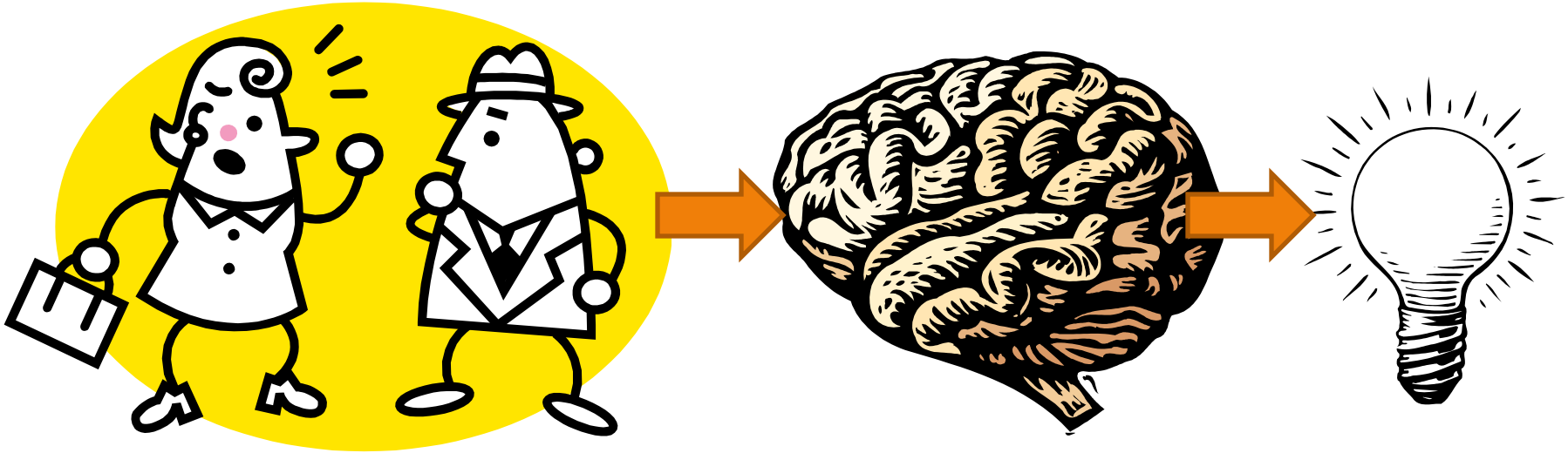
Background: digitalblasphemy.com

# Connecting the Dots

- Cognitive Science has discovered lots about how the brain computes language
- We want to write down how a particular language is structured so we can run computationally simulate understanding of sentences in the language
  - First we need a theory of how language is learned & structured. **Construction Grammar** is one such theory.
  - **Embodied Construction Grammar** (ECG) is a formalism developed to facilitate computational analysis and simulation of linguistic input

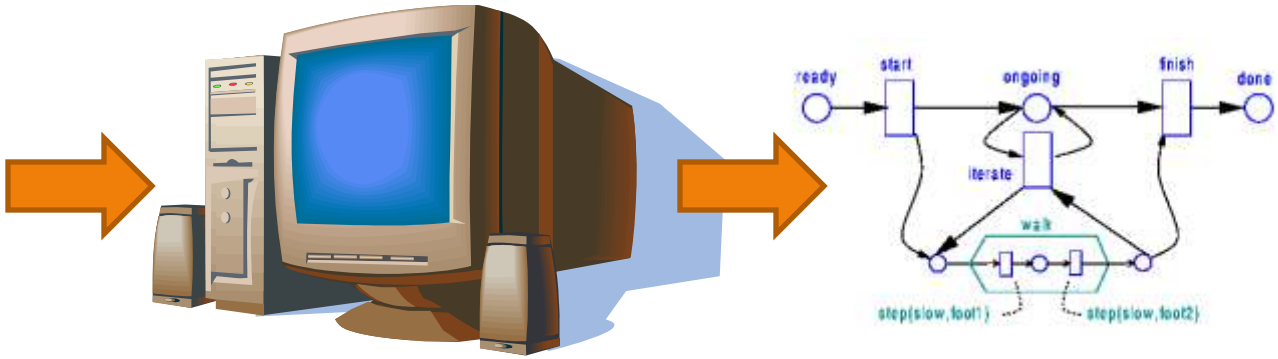
# Connecting the Dots

We want to simulate



with

**Blah blah blah.**



# Outline

- Background: Traditional theories in linguistics
- Construction Grammar (overview)
- Embodied Construction Grammar
  - Overview
  - Schemas
  - Constructions
  - Semspecs
  - Simulation

# Traditional Theories in Linguistics

- Linguistics since the mid-20th century has been dominated by approaches & assumptions advanced by Noam Chomsky
- These theories variously known *generative*, *formal*, *Chomskyan grammar*. Henceforth: "Generative"
- Linguists at Berkeley (and elsewhere on the West Coast) have been among the most vocal critics of some aspects of these approaches

# Generative Philosophy

- Human language is best studied as a formal, deterministic symbolic system, much like math or programming languages.
  - "Language is a self-contained system amenable to algorithmic characterization, with sufficient autonomy to be studied in essential isolation from broader cognitive concerns" (Langacker 2002).

# Generative Philosophy

- Two distinct components of language: *lexicon* and *grammar*
  - **Lexicon:** dictionary of words/meaningful word parts. All "unpredictable" (noncompositional) constructions must be listed here.
  - **Grammar:** rules about how structures are formed by combining lexical entries.
    - **Productivity:** A rule is *productive* if it can apply to novel forms.
    - **Compositionality:** A *compositional* word or phrase has a predictable meaning given its parts.

# Generative Philosophy

- Grammar is composed of several autonomous "modules" which operate in sequence to produce/interpret a sentence
  - **Phonology** – How sounds combine to form *word parts* (*morphemes*—roots, prefixes, suffixes, etc.)
  - **Morphology** – How morphemes combine to form *words*
  - **Syntax** – How words combine to form *phrases and sentences*

# Generative Philosophy

- **Semantics** is the truth-conditional meanings conveyed by a sentence, and **pragmatics** is how sentences are interpreted in context. These are independent of grammar.
- Chomksy himself, and the bulk of the field, have focused their attention on syntax.

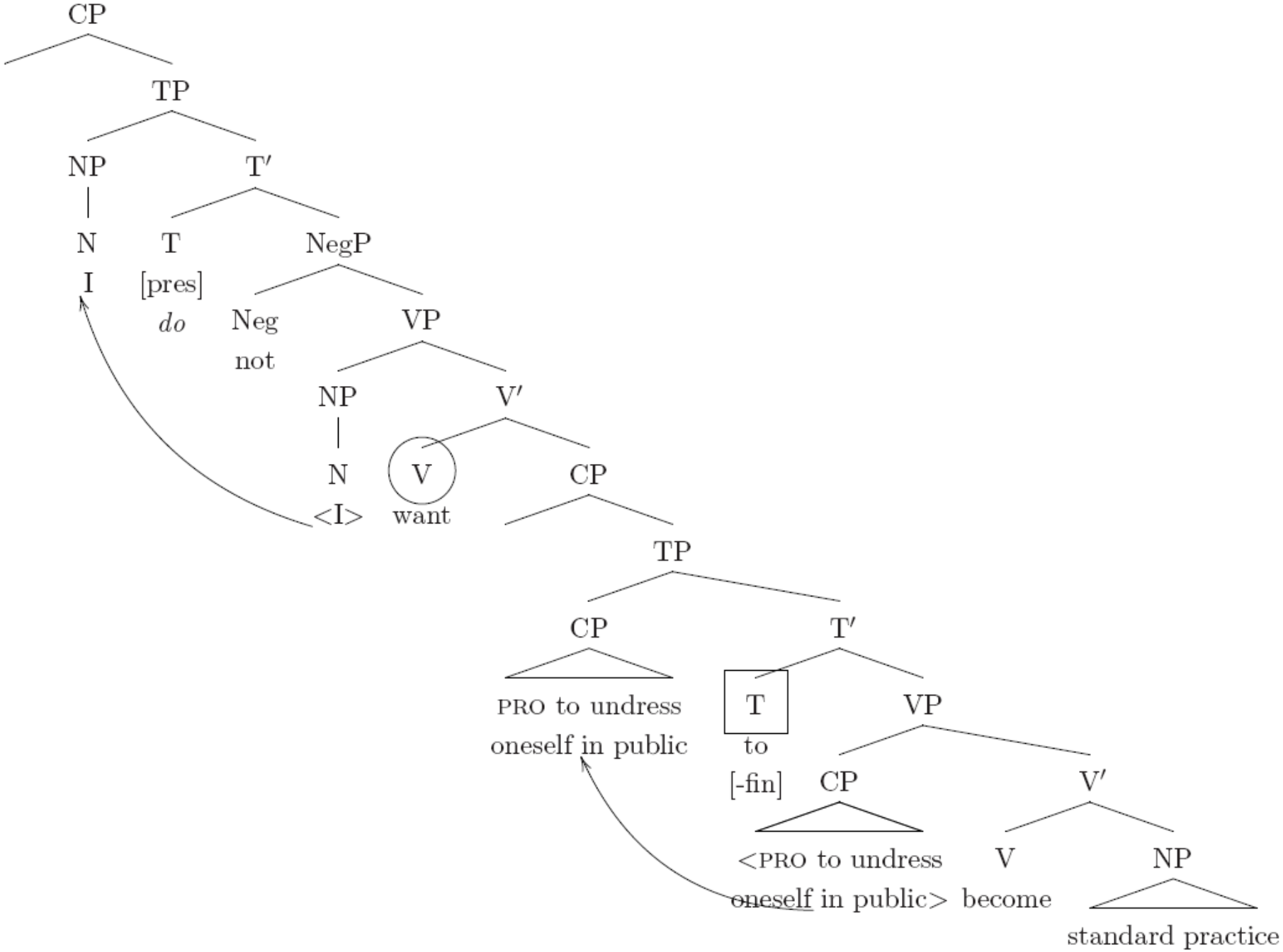
# Generative Philosophy

- The study of syntax is the determination of what abstract structures and processes suffice to characterize grammatical sentences in the language, while excluding ungrammatical sentences.
  - Various stages of Chomsky's syntactic theories include *Transformational Grammar*, *Minimalism*, *X-bar Theory*. These are similar in spirit to context-free grammars with some additional machinery on top: they posit that each sentence has an **underlying representation** which is then manipulated by a series of operations to produce the **surface representation** (with the word order that is actually observed).

# Generative Philosophy: Syntax

- Generative theories of syntax grow increasingly baroque as more and more types of English sentences turn out to be grammatical.

Figure 1: I do not want to undress oneself in public to become standard practice.



# Competence vs. Performance

- The **competence vs. performance** distinction: Models of linguistic *competence* claim only to account for the output (sentences) given the input (communicative intent). They say nothing about what's going on in the human brain when using language (*performance*).
  - Many (if not most) linguists are interested in developing models describing linguistic output without regard for psychological or neural factors governing actual language use.
  - The argument of this class: This is not the most interesting problem! **Language is a human phenomenon; thus to truly understand it, we should endeavor to understand how it works in humans.**

# Generative Philosophy: Universal Grammar

- Chomsky has a famous theory called the **Universal Grammar hypothesis**. His argument: Language is way too complex for children to learn in a couple short years based on what they hear ("poverty of the stimulus")
  - They get incomplete information (there are infinite possible sentences, they hear only a few)
  - Children have no ungrammatical examples to help them refine their models

# Generative Philosophy: Universal Grammar

- Thus, most *principles* of language must be **innate**, i.e. prespecified in the brain. All the child has to do is tune a few *parameters*, settings which serve to account for all the variation in the world's languages.
- Any normal child can learn any language if sufficiently exposed to it; thus this innate knowledge about what principles underlie language must be **universal**.
- This innate linguistic knowledge is assumed to reside in a particular part of the human brain reserved for language—a "language module."
- Based on what you've learned in this class, what problems do you see with this hypothesis?

# Refuting Generative Philosophy

- **Universal Grammar** is an untenable theory based on what we know about biology and embodiment
  - The brain evolved to run the body.
  - Language is so powerful because it rests on existing perceptual and conceptual structures that we use to understand and interact with the world.
  - The brain is massively connected, so it's naïve to think there's a language "organ" which doesn't play a role in other cognitive capacities.
- Argument against "poverty of the stimulus": Children have access to vast semantic knowledge that guides them in their understanding and learning of language ("opulence of the substrate").

# Refuting Generative Philosophy

- A few other problems with Generative philosophy:
  - The different "layers" of language—semantics, syntax, morphology, phonology—do, in fact, interact.
  - It is impossible to completely separate the grammar from the lexicon.
  - The brain is a stochastic system, unlike a computer. It does not involve symbol-manipulation, but rather neural computation/spreading activation. So syntax (and the rest of grammar) uses *probabilistic best-fit*, not deterministic rules.

# Idioms: Why Generative Grammar Fails

- An interesting case to consider are English idioms like "kick the bucket."
  - Ordinary words in ordinary syntax, but a special (idiosyncratic) meaning
  - A Generative approach might simply relegate idioms to the lexicon, which fails to capture that they have ordinary structure with extraordinary meaning

# Idioms: Why Generative Grammar Fails

- Another class of idiomatic constructions are partially productive: they have "slots" that can be filled in with certain types of words.
  - "What's **X** doing **Y**?" (e.g. "What's that fly doing in my soup?") This uses ordinary syntax, but the idiomatic sense means something like "Why is it that X is Y?!"
  - "The **X**-er, the **Y**-er." (e.g. "The more, the merrier"; "The easier it is, the faster I'll finish it.") This uses unusual syntax, and asserts a correlation between two scales against which X and Y are measured.
  - These constructions can't be entirely in the lexicon, because parts of them can vary within some category. But they don't follow the "normal" rules of compositionality, which means a Generative view of syntax cannot account for them.

# Outline

- Background: Traditional theories in linguistics
- Construction Grammar (overview)
- Embodied Construction Grammar
  - Overview
  - Schemas
  - Constructions
  - Semspecs
  - Simulation

# Construction Grammar

- A radical new(-ish) approach to a theory of language is called *Construction Grammar* (CxG)
- CxG refers to a family of theories of grammar which share some basic assumptions
- A few names associated with CxG:

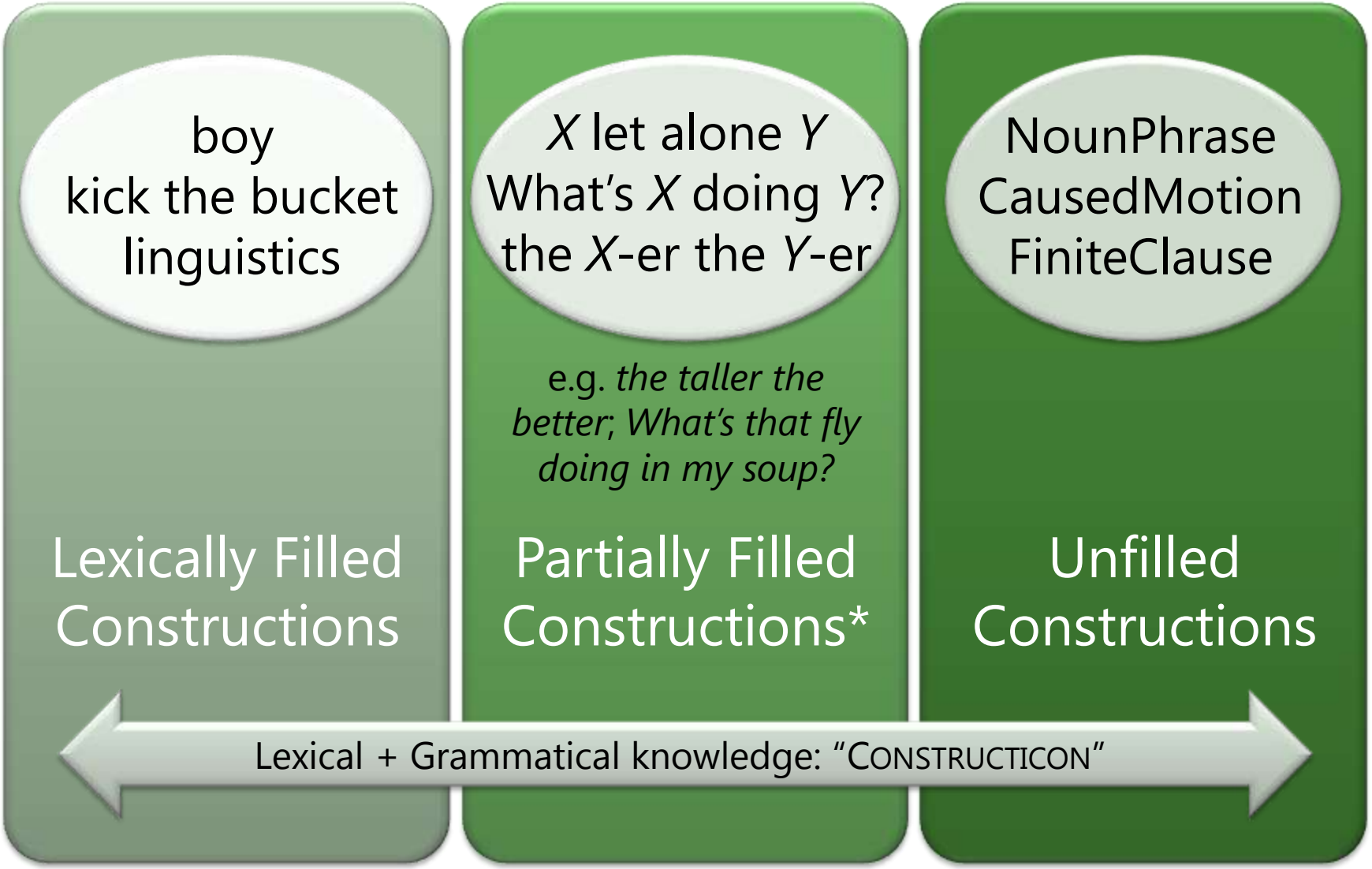
| | | |
|---|---|---|
| **Charles Fillmore** | Emily Bender | Joan Bybee |
| **Paul Kay** | Ben Bergen | William Croft |
| **George Lakoff** | Adele Goldberg | Ronald Langacker |
| Karen Sullivan | Mirjam Fried | Michael Tomasello |
| Eve Sweetser | Laura Michaelis | Arnold Zwicky |
| (here!) | (spent time here!) | |

# Construction Grammar

The fundamental units of grammar/linguistic knowledge are *constructions*, which are conventionalized pairings of **form** and **meaning**.

- A construction (cxn) can be as simple as a word or as complex as a phrasal pattern
- "*Form* in constructions may refer to any combination of syntactic, morphological, or prosodic patterns and *meaning* is understood in a broad sense that includes lexical semantics, pragmatics, and discourse structure" (Fried)
- Learning a language means **learning its constructions** and how they combine to convey meaning. "It's constructions all the way down" (Goldberg 2005)
- **Meaning** is crucial at all levels of grammar

# Construction Grammar: Constructicon

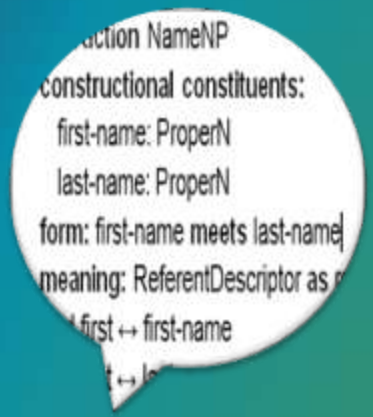



* (Fillmore, Kay, & O'Connor 1988; Kay & Fillmore 1999)

# Construction Grammar: Levels


Form

Phonology




Constructions (form+meaning)

Morphology & Syntax

Meaning

Semantics & Pragmatics

(e.g. frames)

Linguistic Structure

# Construction Grammar: Useful Evidence

- Psycholinguistic and neural evidence is crucial to understanding how constructions are learned and used
  - **Usage-based models:** Learning is done through **bottom-up generalizations**. Initially the small pieces (words) are learned, then generalizations are made about word order (phrasal constructions). Semantics (much of which the child can infer from context) plays a critical role in learning.
  - **Frequency information** about constructions is learned (think: neural connections strengthened with use) and plays an important role in linguistic phenomena.

# Construction Grammar

- In a *usage-based* model of grammar (Goldberg 2006), one learns individual constructions and generalizes bottom-up to form new constructions, all of which can coexist
  - General constructions are said to *motivate* the form/meaning of more specific constructions
    - A general NounPhrase construction might motivate special cases like Determiner-Noun and Adjective-Noun
  - Idiosyncrasy is handled gracefully: specific constructions can override or augment generalizations

# Construction Grammar

- **Non-derivational:** Unlike in Generative syntax, the observed representation is not somehow a permutation of another representation.
  - Paraphrases (e.g. active vs. passive sentences) nevertheless have different discourse properties, pragmatics, or entailments—all parts of meaning
  - Truth-conditional semantics is insufficient to capture the richness of all that is communicated
- What's the difference between these sentences?: [with apologies to (Goldberg 1995)]

  1) Sam baked a cake for Amy.
  2) Sam baked Amy a cake.

# Construction Grammar

- Amy had promised to bring food to the symposium but got sick and couldn't attend, so…
  3) …Sam baked a cake for Amy (on her behalf).
  4) …*Sam baked Amy a cake.
- The English ditransitive denotes (possibly metaphorical) *transfer*, and the indirect object is an intended recipient. An approach which proposes to derive (4) from (3) will not capture this. Similar arguments apply to passives, etc.

# Construction Grammar

- Constructional compositionality: A construction's constituent constructions can contribute to its meaning, but the whole can be greater than the sum of the parts

5 4 3 2 1

# Outline

- Background: Traditional theories in linguistics
- Construction Grammar (overview)
- **Embodied Construction Grammar**
  - **Overview**
  - Schemas
  - Constructions
  - Semspecs
  - Simulation

# ECG: Goals

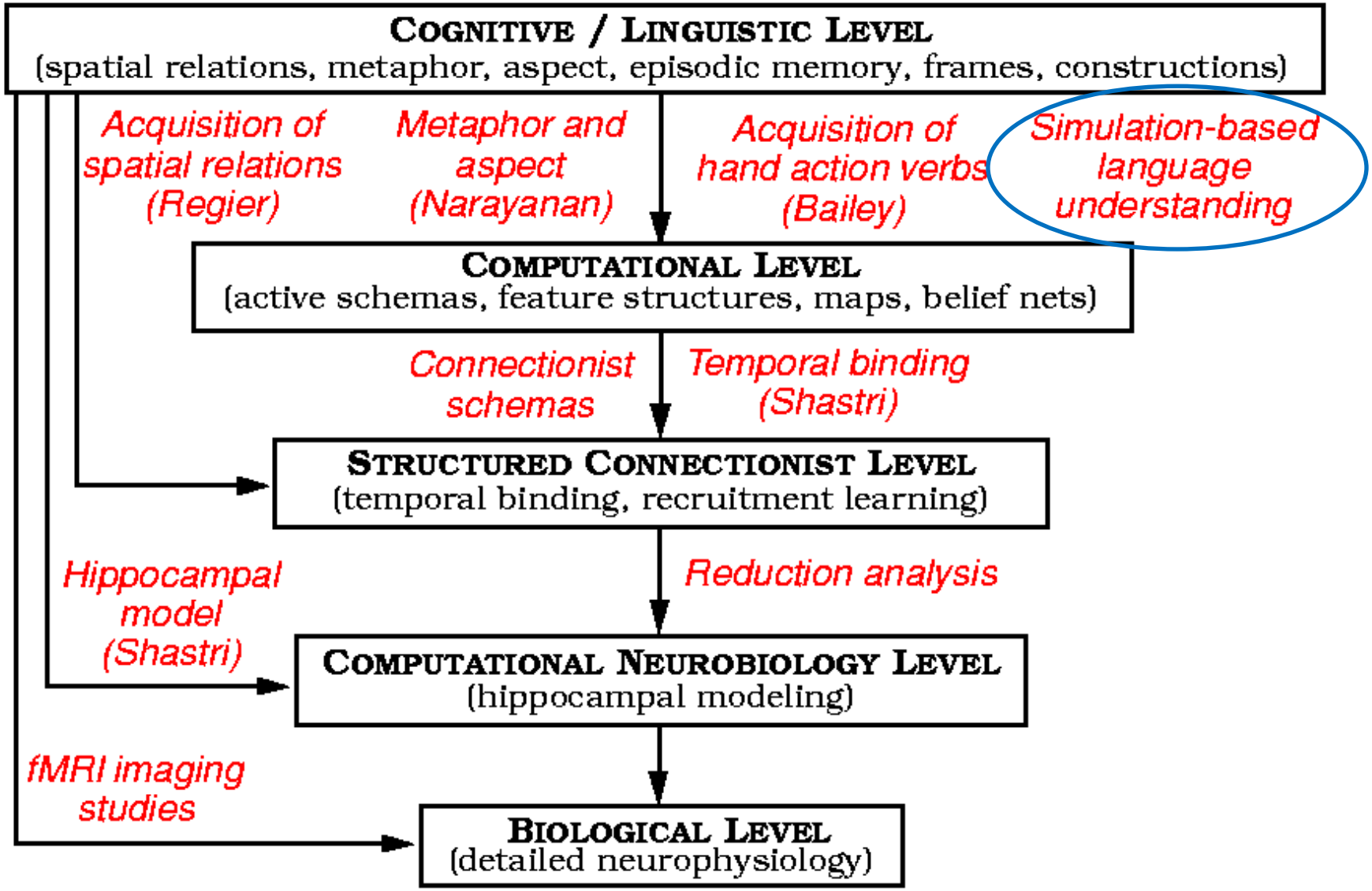
- Embodied Construction Grammar (ECG)*: a project of the Neural Theory of Language group here at Berkeley
- It is a **formalism** for representing constructions in a language by mapping form to (frame-based) meaning
  - A precise, computationally verifiable notation for **linguistic analysis**
  - A bridge from **sentences** to **simulations**

* Ben Bergen & Nancy Chang (2005). Embodied Construction Grammar in simulation-based language understanding.

# ECG: Goals

- ECG is a *high-level symbolic* representation for grammars, thought to be mappable to a *low-level neural* representation.
- The NTL relies heavily on two principles:
  - **Best-Fit Hypothesis:** The choice of constructions in production is based on a probabilistic best fit of the inventory of stored constructions to match the communicative intent; likewise, the hearer's interpretation is based on a best choice of constructions given the utterance and context.
  - **Simulation Hypothesis:** Understanding is mental simulation. (think: mirror neurons)

# NTL: Levels of Abstraction



# NTL People who work on ECG



*Professors*

**Jerry Feldman** CS

**George Lakoff** Ling/CogSci

**Eve Sweetser** Ling/CogSci


*Grad Students*

**Eva Mok** CS

**Johno Bryant** CS

**Ellen Dodge** Ling

http://www.icsi.berkeley.edu/NTL/people.php

# Language Understanding: Two Phases

**construction** WALKED
**form**
**self.**$_f$.phon ← [wakt]
**meaning :** Walk-Action
**constraints**
**self.**$_m$.time *before* Context.speech-time
**self.**$_m$..aspect ← encapsulated

"Harry walked into the cafe."

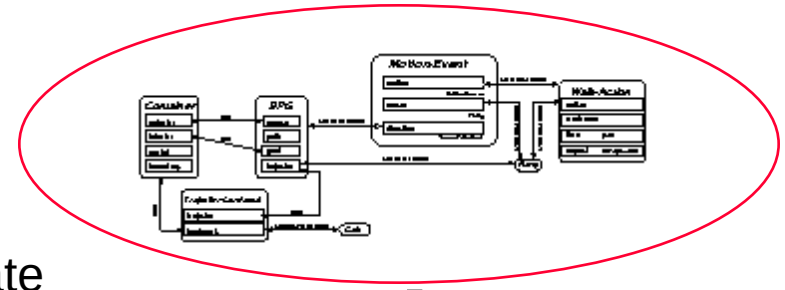
Utterance

Constructions

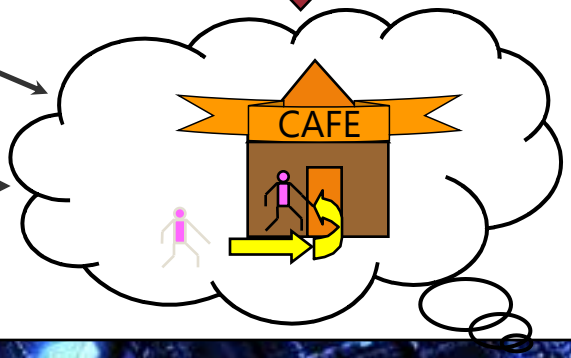
**Analysis Process**

General Knowledge

Belief State



Semantic Specification (Semspec)




**Simulation**

# Structures from Cognitive Linguistics

The ECG **primitives** are types of objects corresponding to key cognitive structures:

- **Schemas** (/frames)
  - Image schemas, force dynamics, executing schemas…
- **Constructions**
  - Lexical, grammatical, morphological, gestural…
- **Maps**
  - Metaphor, metonymy, mental space maps…
- **Mental spaces**
  - Discourse, hypothetical, counterfactual…

# The Definition vs. Instance Distinction

- As is standard in object oriented languages, ECG distinguishes between:
  - An object (schema or construction) **definition/type**, and
  - An object **instance** created according to the definition.
- There may simultaneously be multiple instantiations of a single type
- Definitions comprise the **grammar**; instances are bound together to form the **Semspec**

# Binding/Unification

- ECG belongs to a class of grammars known as *unification grammars*
  - HPSG is another unification grammar
- Basic idea: To represent language we need to instantiate a bunch of structures (constructions and schemas) that are linked together via roles
  - There are (type, value, identity) constraints on what can be linked to what
  - **Unification** = the process of constrained linking. In ECG we call it **binding** because of what it represents neurally.

# Outline

- Background: Traditional theories in linguistics
- Construction Grammar (overview)
- **Embodied Construction Grammar**
  - Overview
  - **Schemas**
  - Constructions
  - Semspecs
  - Simulation

# ECG Schemas: Three Types

- There are three types of schemas: *semantic schemas*, *form schemas*, and *constructional schemas*
  - **Semantic schemas** define the internal structure of a concept. This includes frames as well as universal schemas. We will focus on these.
  - **Form schemas** such as Word define a structured representation of linguistic form (e.g. phonology, orthography)
  - **Constructional schemas** define grammatical attributes of constructions (think morpho-syntactic features like gender and number)

# Schema Formalism

**schema** *Name*

**subcase of** *SuperType1, …*

**evokes** *SchemaType* **as** *localName*

**roles**

*role1*: *Type1*

*role2*: *Type2*

**constraints**

*roleA* ← *literalValue*

*roleB* ↔ *roleC*

**ignore** *roleD* ↔ *roleE*

**Schema**: Defines a new schema type

**Subcase of**: Asserts that this schema is a subtype of one or more other schema types, and should inherit their properties

**Evokes**: Used to relate this schema to an instance of other schema, which may already be present in the analysis

**Schema roles**: Entities/events that the schema relates. They are filled by schema instances, which must conform to the (optional) type specifications.

**Literal assignment**: A constant value (such as a string or numeric) is specified

**Binding/unification**: Asserts that the filler objects referred to by these two roles are one and the same

**Ignore**: Nullifies an inherited constraint

# Evokes

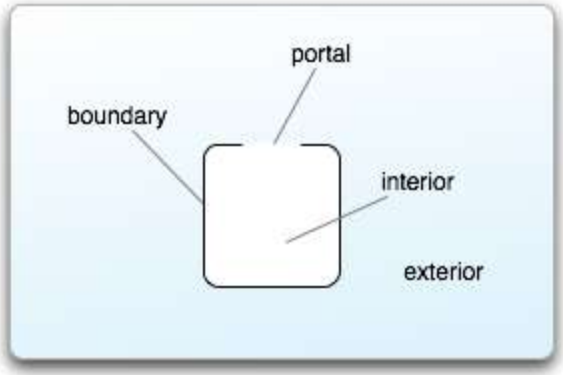
- The **evokes** statement is unique to ECG. It specifies a schema type to be evoked and local identifier to the evoked instance.
  - Procedure: The analyzer searches the schema instances that have been created so far in the analysis to see if one matches the evoked type and any constraints imposed on the evoked instance. If no such schema exists, a new instance of the evoked type is created.
  - What neural process does this model?
    - Spreading Activation

# Evokes

- Example: Recall that 'hypotenuse' is a concept which is defined relative to a RightTriangle frame. So the Hypotenuse schema would **evoke** the RightTriangle schema.
  - "This right triangle has a **hypotenuse** of length 5." The evoked schema has been created already for the phrase "right triangle," so the semantic interpretation "hypotenuse" references the existing RightTriangle instance.
  - "This **hypotenuse** is of length 5." No other word or phrase in the sentence instantiates the RightTriangle schema, so the evocation in Hypotenuse will create a new RightTriangle instance.

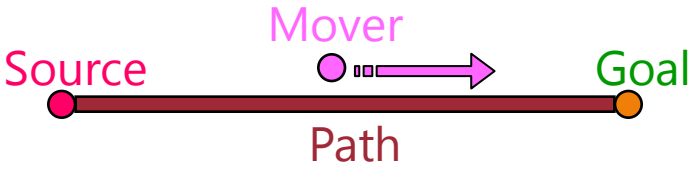
# Some Schemas You've Seen Before

**schema** Container
  **roles**
    boundary
    portal
    interior
    exterior




**schema** TrajectorLandmark
  **roles**
    trajector
    landmark

**schema** SPG
  **roles**
    mover
    source
    path
    goal



# The Into Schema

- The Into schema binds these in the configuration that characterizes the meaning of 'into'. It is a type of TrajectorLandmark relation, so the trajector and landmark roles are implicit in the definition. (If Into needed additional roles, they would be defined in a **roles** block.)

```
schema Into
    subcase of TrajectorLandmark
    evokes Container as cont
    evokes SPG as spg
    constraints
        landmark ↔ cont
        spg.source ↔ cont.exterior
        spg.path ↔ cont.portal
        spg.goal ↔ cont.interior
```

# Outline

- Background: Traditional theories in linguistics
- Construction Grammar (overview)
- **Embodied Construction Grammar**
  - Overview
  - Schemas
  - **Constructions**
  - Semspecs
  - Simulation

# ECG Construction Formalism

**construction** *Name*
  **subcase of** *SuperType1, …*
  **constructional**: *CxnlBlockType*
    **constituents**
      *constit1*: *CxnType1*
    **constraints**
      *constitA* ↔ *constitB*
      *cxnlRoleA* ← *literal*
  **form**: *FormBlockType*
    **constraints**
      *fRoleA* ← *literal*
      *constitA*.f **before** *constitB*.f
      *constitC*.f **meets** *constitD*.f
  **meaning**: *SchemaType*
    *// Anything that can go in a schema definition can go here (except* **subcase***).*
    *// Constraints that refer to constituents' meanings must have constitName*.m

**Construction**: Defines a new cxn type.

**Subcase of**: Asserts that this cxn is a subtype of one or more other cxn types, and should inherit their properties.

**Constructional block**: Pertains to the grammatical attributes of the cxn. Constituents are component constructions. May have a constructional schema type such as NounFeatures.

**Form block**: Specifies the linguistic form of the cxn. May have a form schema type such as Word which defines properties for wordforms (such as orthography).

**Before constraint**: Asserts that a constituent must precede another in the input.

**Meets constraint**: Asserts that a constituent must *immediately* precede another in the input.

# Notes on ECG Syntax

- All three blocks of a cxn—**constructional**, **form**, **meaning**—are optional. If present, they need to be defined in that order.
- Type declarations are optional and introduced by a colon. Untyped block/role declarations should not have a colon.
- Constructional constituents are members of the cxn object which can be accessed from either the form or meaning block. The form block must refer to the constituent's form pole using .f , and the meaning block to the meaning pole using .m . Likewise for the **self** keyword.

# The IntoCxn Construction

- Recall the Into schema:

```
schema Into
    subcase of TrajectorLandmark
    evokes Container as cont
    evokes SPG as spg
    constraints
        landmark ↔ cont
        spg.source ↔ cont.exterior
        spg.path ↔ cont.portal
        spg.goal ↔ cont.interior
```

- The IntoCxn* construction associates this semantic schema with its English linguistic (here, written) form:

```
construction IntoCxn
    subcase of SpatialPreposition
    form: Word
        orth ← "into"        // orthography
    meaning: Into
```

* The Cxn in IntoCxn is there to disambiguate the schema and construction type names.

# The IntoCxn Construction

- Equivalently, we could specify all of Into's semantics directly in the construction definition (without a special Into schema):

```
construction IntoCxn
    subcase of SpatialPreposition
    form: Word
        orth ← "into"        // orthography
    meaning: TrajectorLandmark
        evokes Container as cont
        evokes SPG as spg
        constraints
            landmark ↔ cont
            spg.source ↔ cont.exterior
            spg.path ↔ cont.portal
            spg.goal ↔ cont.interior
```

# Phrasal Constructions

- IntoCxn is called a *lexical construction* because it refers to a single word
- SpatialPP* is an example of a phrasal (or syntactic) construction because it combines the forms and meanings of multiple lexical constructions:

```
construction SpatialPP
    subcase of Phrase
    constructional constituents
        prep: SpatialPreposition
        lm: ReferringExpression
    form constraints
        prep.f before lm.f
    meaning constraints
        prep.m.landmark ↔ lm
```

* PP = prepositional phrase

# General Constructions

- Many constructions are generalizations over other constructions
- We often wish to define *general* (or *abstract*) constructions that are never instantiated on their own, but encapsulate shared properties of several *concrete* construction subtypes
  - ECG provides the **general** keyword for this purpose

# General Constructions

- For instance, it is useful to have a single construction capturing the shared properties of all spatial prepositions:

```
general construction SpatialPreposition
    subcase of Preposition
    meaning: TrajectorLandmark
        // etc.
```

- We never want to be able to instantiate <u>SpatialPreposition</u> directly, though—it's too vague! Hence, it is defined as **general**.
  - It can be used in type declarations
  - A concrete descendent can be instantiated outright

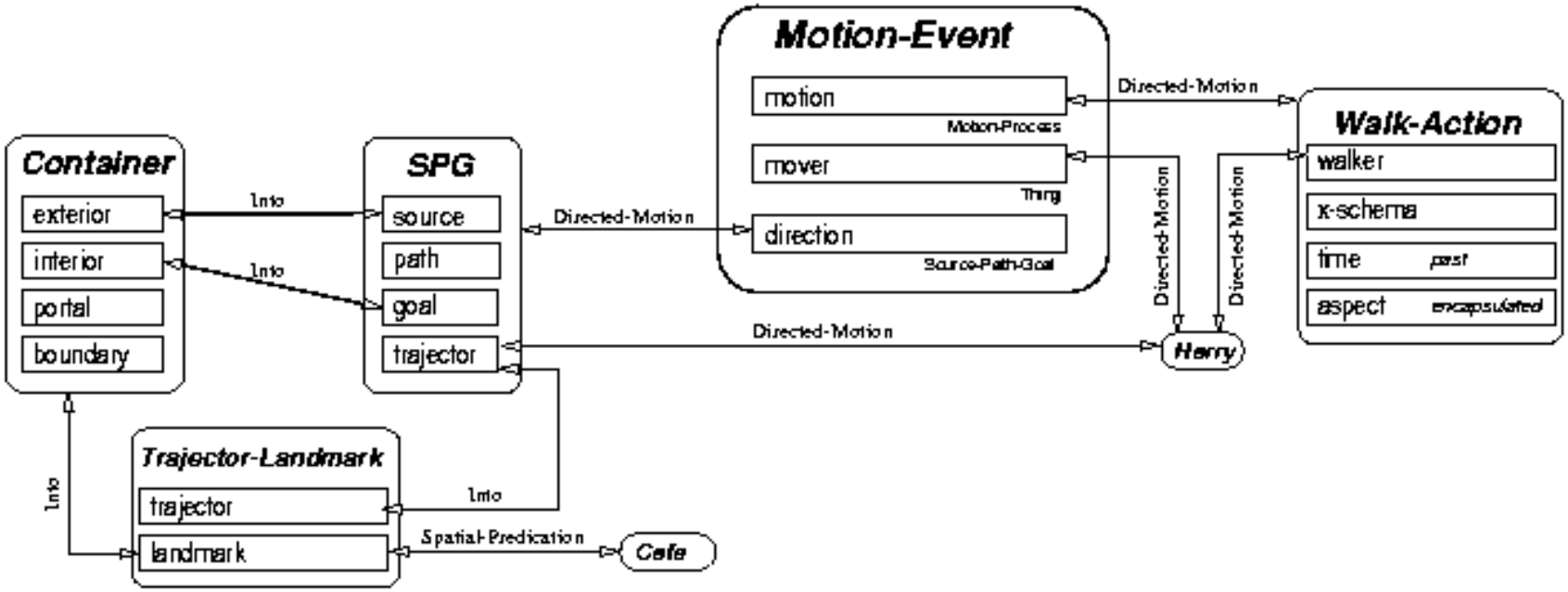
# Outline

- Background: Traditional theories in linguistics
- Construction Grammar (overview)
- **Embodied Construction Grammar**
  - Overview
  - Schemas
  - Constructions
  - **Semspecs**
  - Simulation

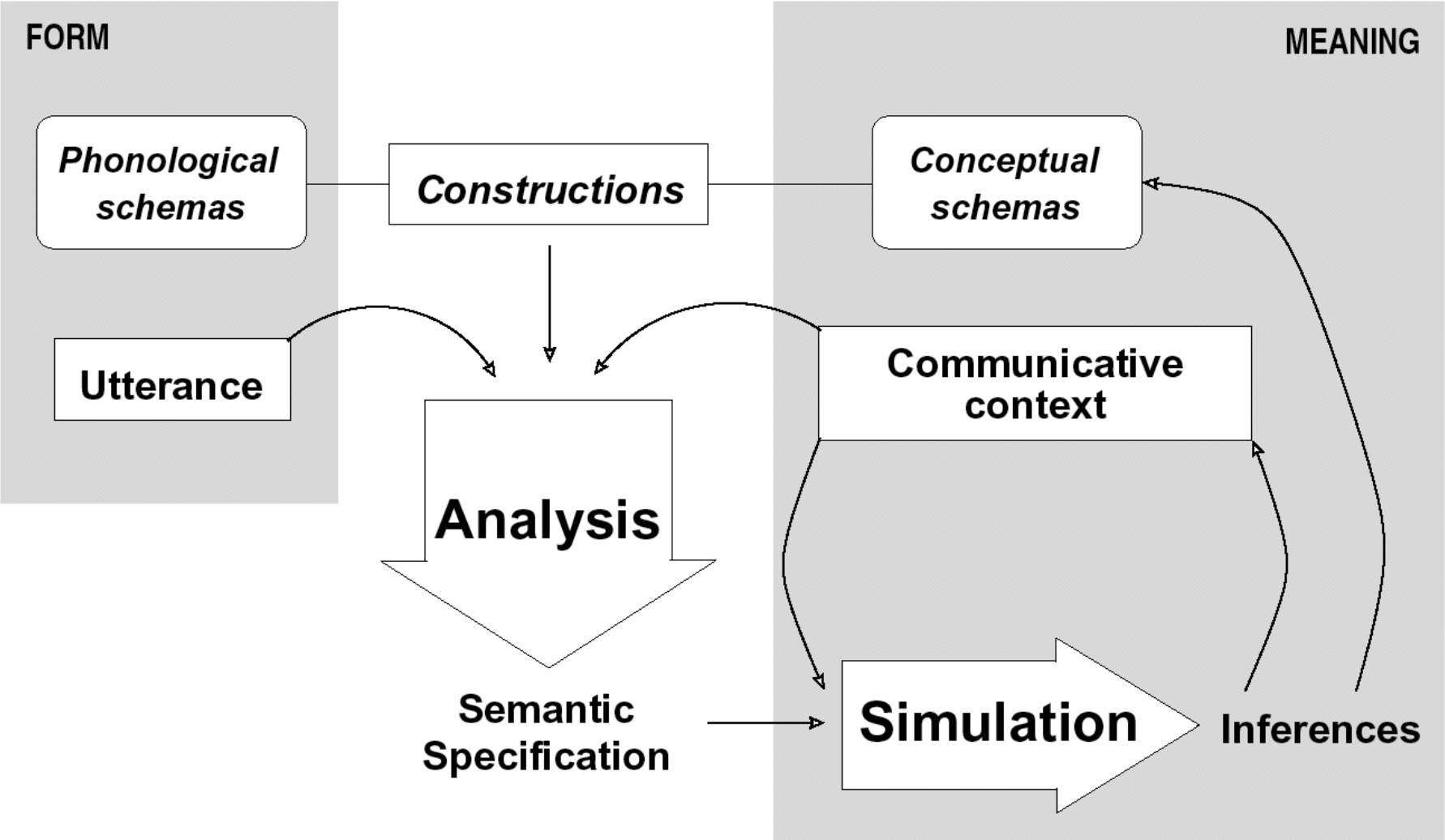
# The Semantic Specification (Semspec)

The analysis process produces a **semantic specification** that

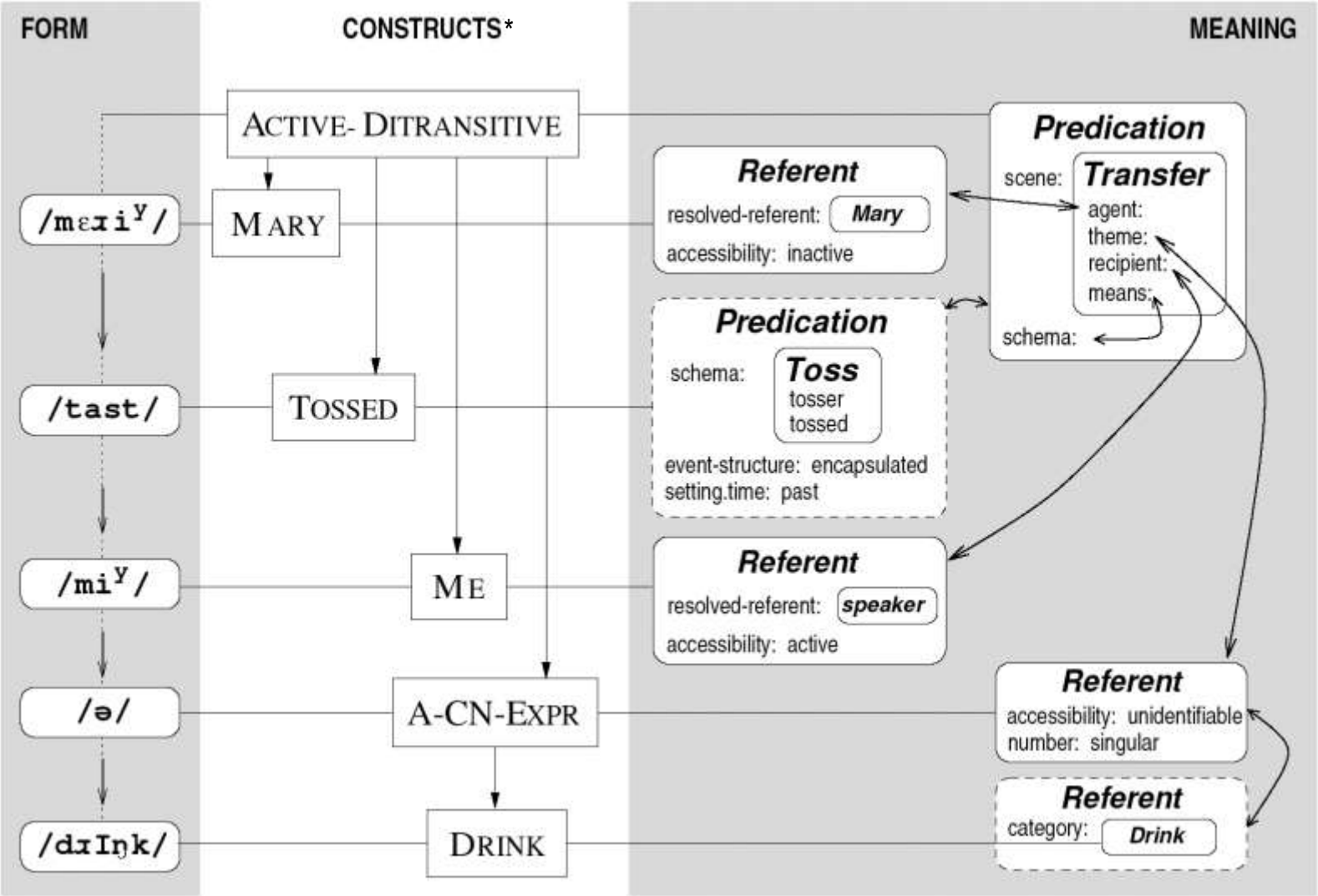
- Includes image-schematic, motor control, and conceptual structures
- Provides parameters for a mental simulation



# Language Understanding Process

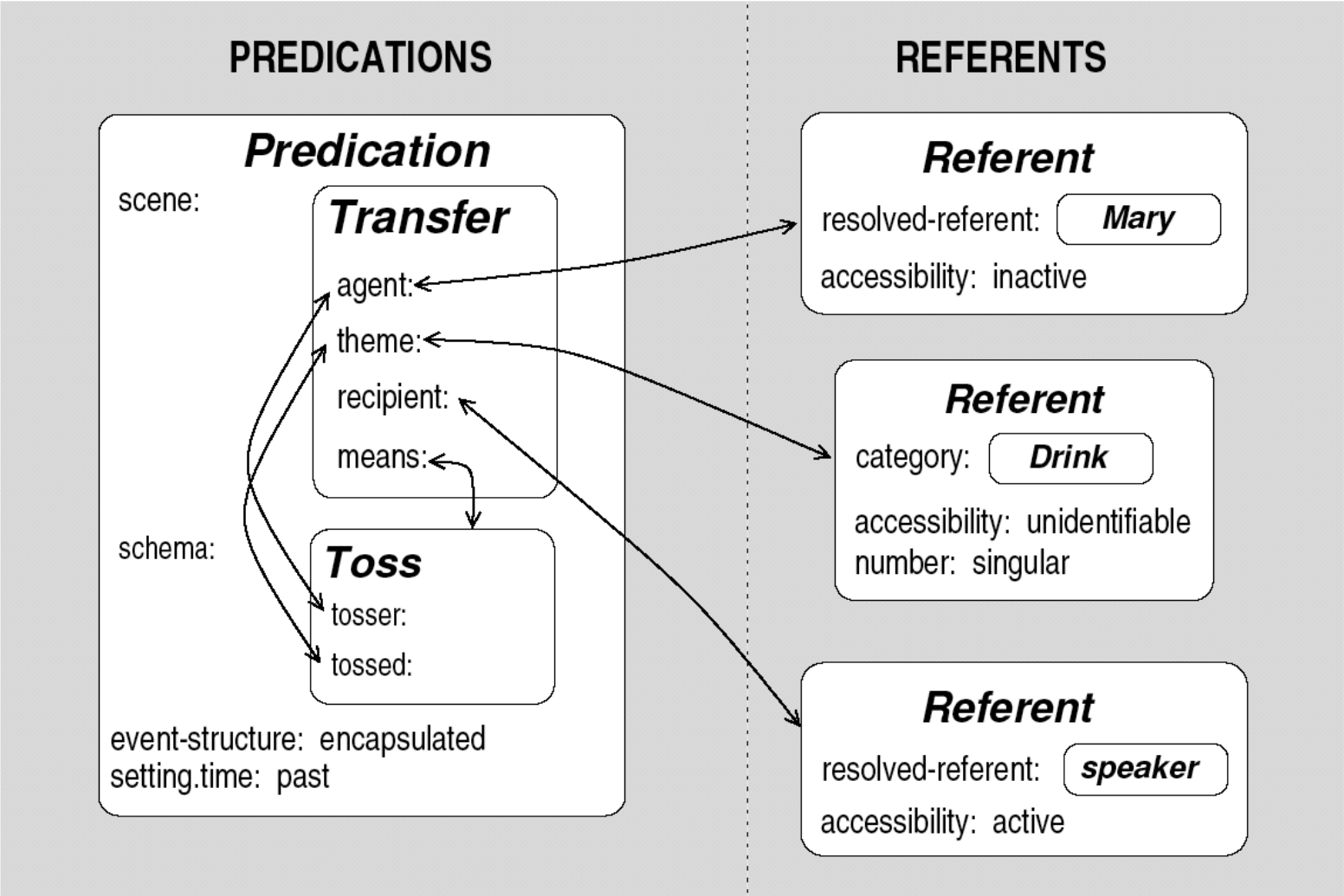


# Constructional Analysis




* Construct = construction instance

# Semantic Specification



# Coming Attractions

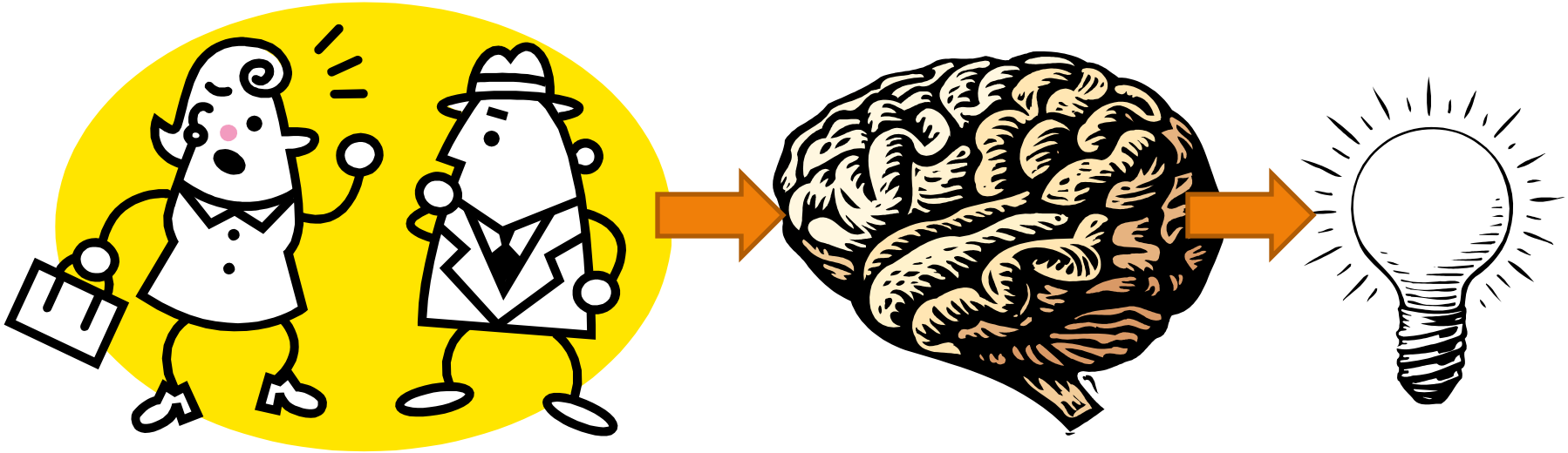
- Background: Traditional theories in linguistics
- Construction Grammar (overview)
- **Embodied Construction Grammar**
  - Overview
  - Schemas
  - Constructions
  - **Semspecs – How they are computed!**
  - **Simulation – How it works!**

# ECG Summary

- **Parts** = constructions, schemas. All told, their definitions constitute the grammar.
- **Combination** = binding (unification) of parts (instances). The complete network is the Semspec.
- **Meaning of the whole** = simulation of the combined parts

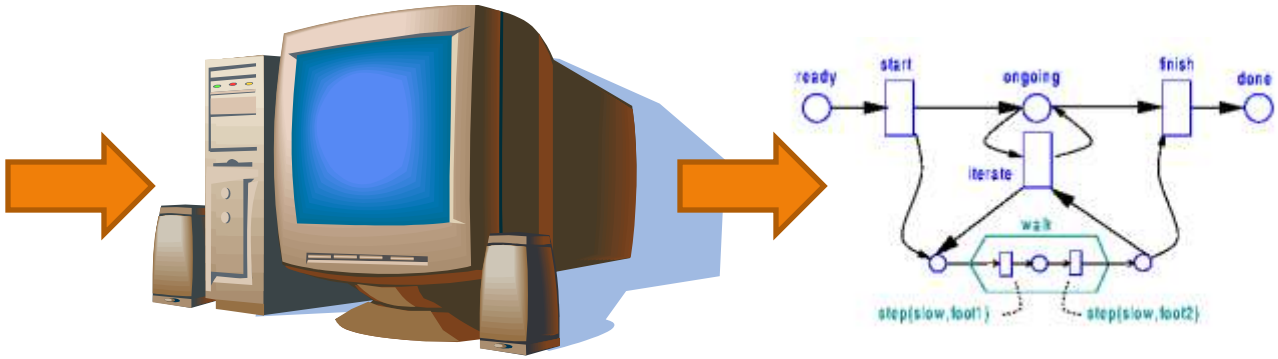
# Connecting the Dots

We want to simulate



with

**Blah blah blah.**



# Bibliography


Bergen, Ben, & Chang, Nancy (2005). Embodied Construction Grammar in simulation-based language understanding. In J.-O. Östman, & M. Fried (Eds.), *Construction grammar(s): Cognitive and cross-language dimensions*. John Benjamins.

Bryant, John (forthcoming). Ph.D. dissertation, University of California, Berkeley.

Feldman, Jerome A. (2006). *From molecule to metaphor: A neural theory of language.* Cambridge, MA: MIT Press.

Fillmore, Charles J., Kay, Paul, & O'Connor, Mary C. (1988). Regularity and idiomaticity in grammatical constructions: the case of 'let alone'. *Language, 64* (3), 501-538.

Goldberg, Adele (1995). *Constructions: A construction grammar approach to argument structure.* Chicago: University of Chicago Press.

Goldberg, Adele (2006). *Constructions at work: The nature of generalization in language.* New York: Oxford University Press.

Kay, Paul, & Fillmore, Charles J. (1999). Grammatical constructions and linguistic generalizations: the What's X doing Y? construction. *Language, 75* (1), 1-33.

Langacker, Ronald W. (2002). *Concept, image, and symbol: The cognitive basis of grammar* (2nd ed.). Berlin: Mouton de Gruyter.