Intro to Grammar

(from an NTL Perspective)

John Bryant

International Computer Science Institute

UC Berkeley

What is a grammar?

- Grammar is the system of a language... It’s important to think of grammar as something that can help you, like a friend.

- Grammar tells the users of a language what choices are possible for the word order of any sentence to allow that sentence to be clear and sensible - that is, to be unambiguous.

- ..."prescriptive grammar," a set of "rules" governing the choice of who and whom, the use of ain’t, and other such matters. Promoted by Jonathan Swift and other literary figures of the 18th century, this approach to language prescribes the "correct" way to use language.

- ..."descriptive grammar," which is the study of the ways humans use systems–particularly syntax and morphology–to communicate through language.

- Therefore grammar acts as his tool to create meaning.

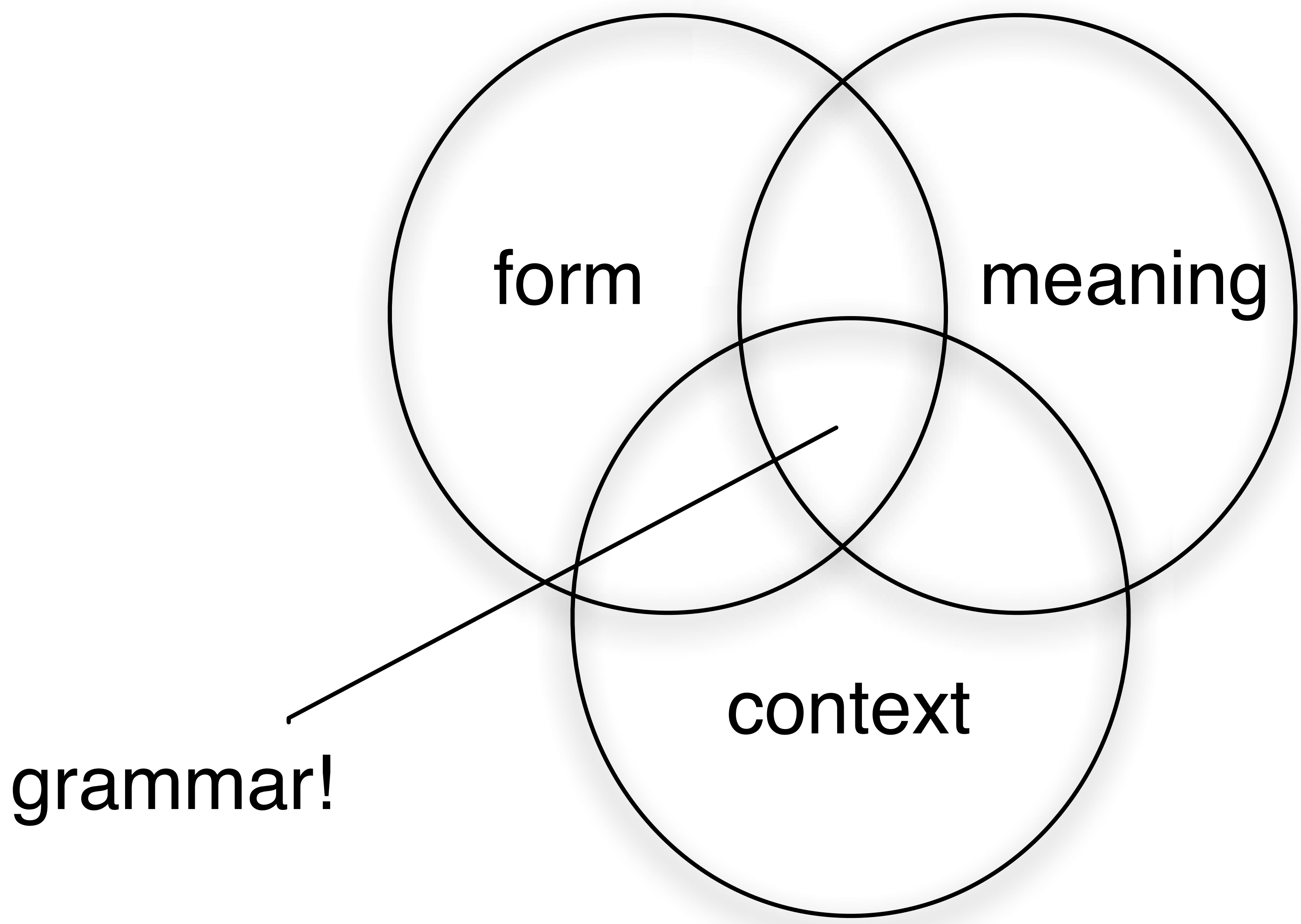
From google

What is an utterance?

- Form
  - Sound
  - Text
  - Sign and Gesture
  - And their order
- Meaning
  - Who doing what to whom (usually more explicit)
  - Pragmatic (usually more implicit)
- Context
  - Shared world state
  - Discourse
  - Ontological knowledge

how do all these pieces fit together?

Grammar is relational knowledge



Embodied Semantics

- Image schemas
- Force dynamic schemas
- Frames
- Metaphors

All these things are bound up and sent to simulation.

Syntax in one slide

Form relations within a sentence, not within a word.

- Word order
- Constituency (grouping and labeling)
- Where constituents can appear
- Grammatical relations (subj, obj)
- Verb subcategorization
- Agreement (number, person, case, gender)

We play the game.	Bill frisbee throws purple the
You am the best!	They hits he.
Bob gave Anne a book.	Anne gave Bob a book.
Tom walked into the cafe.	Tom tumbled into the cafe the ball.
I slept the ball into the basket.	She sneezed the foam off the latte.
This is the man I handed the book	This is the book I handed the man

Formal grammars!

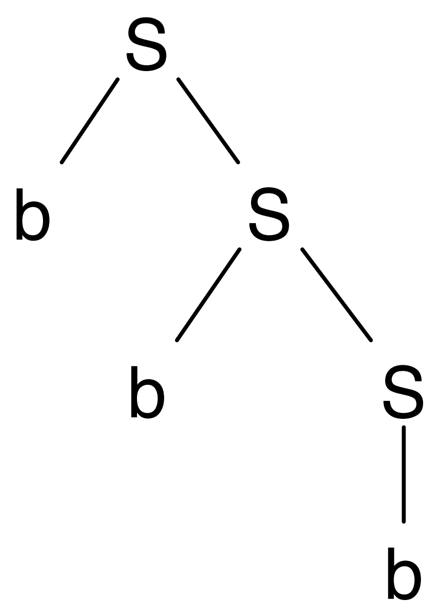
We can:

- Compactly encode a possibly infinite set of sentences.
- Generate a string (sentence) of the language
- Parse (recognize) a string
- Compare the sentences that two grammars can generate.
- Implement stuff
- Make clear, testable predictions about language

$$\begin{array}{llll} 1. & S & \longrightarrow & b\ S \\ 2. & S & \longrightarrow & b \end{array}$$

Parsing a sentence

1. Take an input sentence and a grammar
2. Process the sentence: Top down vs bottom up search
3. Output a trace (parse tree)



1. $S \rightarrow b\ S$
2. $S \rightarrow b$

Formal theories of grammar

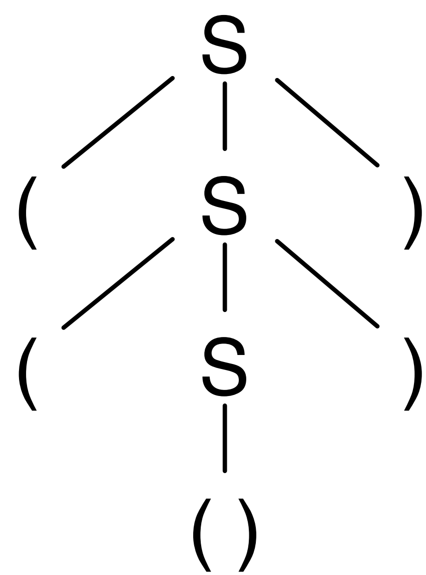
- Finite state grammar
- Context free grammar
- Transformational Grammar, Minimalism, X bar...
- Unification grammar (GPSG, LFG, HPSG)
- Construction grammar (CxG)
- Embodied Construction Grammar (ECG)

Different assumptions about language and different processing complexity.

Context free grammars

A CFG (PSG) is:

- A set of terminals $\subset Symbols$
- A set nonterminals $\subset Symbols$
- A set of productions of the form $NT \rightarrow Symbol*$
- A designated start symbol



A tiny example CFG:

- Terminals: {“(“ , “)”}
- Nonterminals: {S}
- Start symbol = S
- Rules: {S $\rightarrow$ “(” S “)”; S $\rightarrow$ “(“ “)”; }

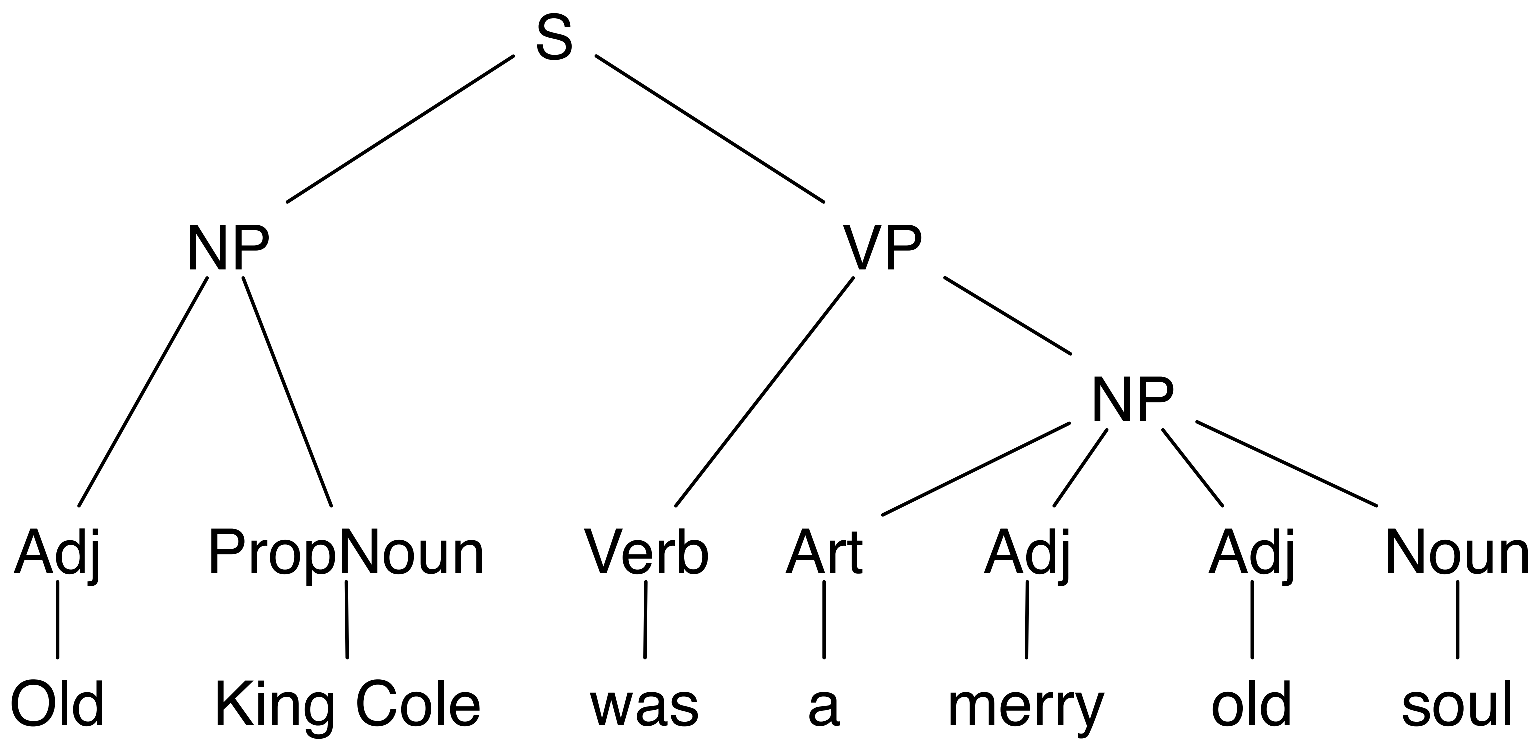
English lexicon fragment

$$
\begin{array}{rcl}
\text{Noun} & \rightarrow & \text{soul} \mid \text{pipe} \mid \text{fiddlers} \mid \text{bowl} \\
\text{ProperNoun} & \rightarrow & \text{King Cole} \\
\text{Pronoun} & \rightarrow & \text{he} \mid \text{they} \mid \text{I} \\
\text{Verb} & \rightarrow & \text{was} \mid \text{called} \mid \text{plays} \mid \text{play} \mid \text{slept} \\
\text{Adjective} & \rightarrow & \text{old} \mid \text{merry} \mid \text{three} \\
\text{Article} & \rightarrow & \text{a} \mid \text{the} \\
\text{Conjunction} & \rightarrow & \text{and}
\end{array}
$$

English grammar fragment

$$
\begin{array}{rcl}
\text{S} & \longrightarrow & \text{NP VP} \\
 & | & \text{S Conjunction S} \\
\\
\text{NP} & \longrightarrow & \text{Adjective ProperNoun} \\
 & | & \text{Possessive Adjective Noun} \\
 & | & \text{Article Adjective* Noun} \\
 & | & \text{Pronoun} \\
\\
\text{VP} & \longrightarrow & \text{Verb NP} \\
 & | & \text{Verb PP} \\
\\
\text{PP} & \longrightarrow & \text{Preposition NP}
\end{array}
$$

An example parse



CFG pros and cons

CFG pros and cons

PROS:

- Simple
- Fast
- Pretty good with word order and constituency

CONS:

- Hard to prevent overgeneration
- No semantics
- Too simple?

Generated sentences

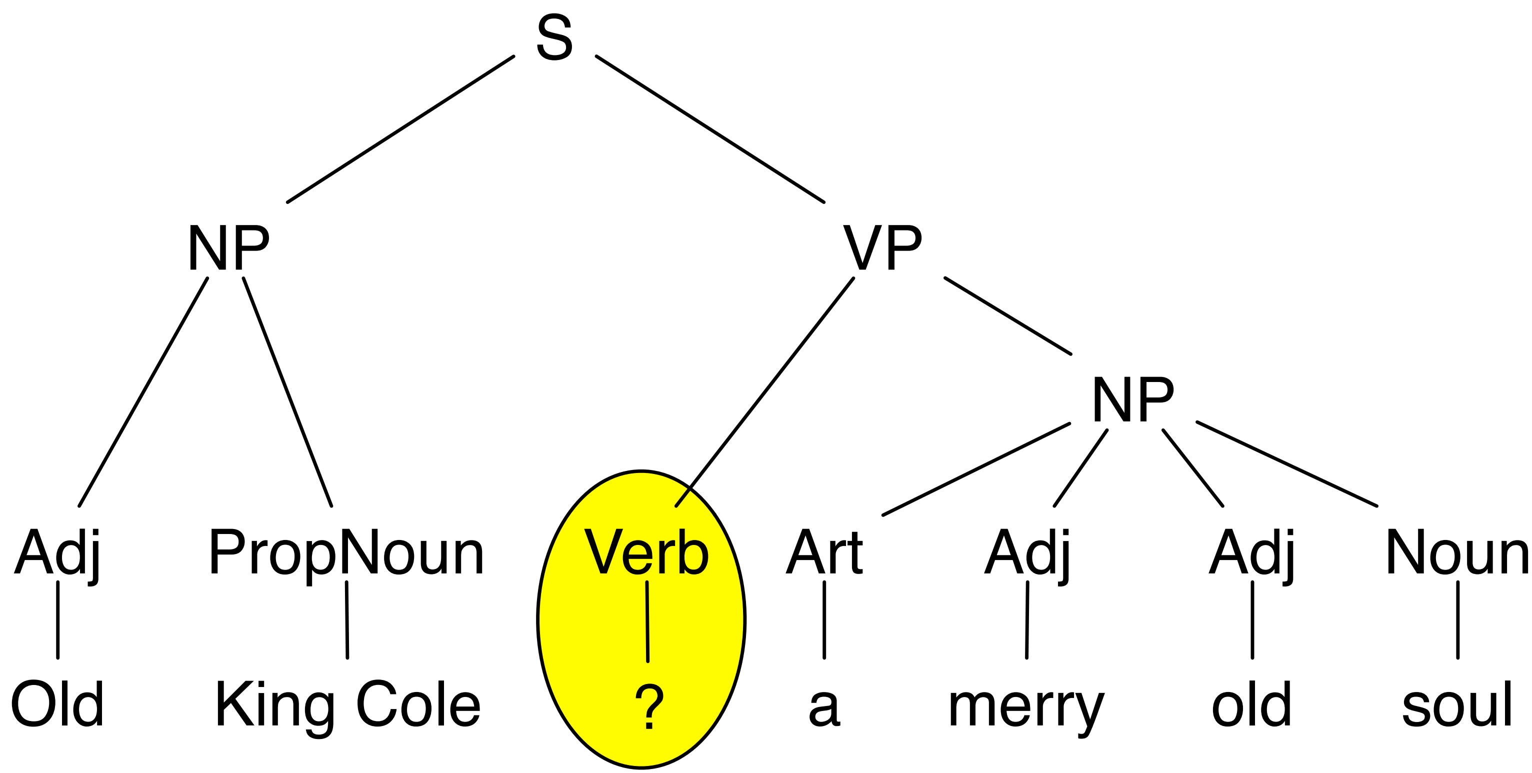
Goodies:

- Old Kind Cole was a merry old soul
- A merry old soul was he
- He called for his pipe
- He called for his bowl
- He called for his three fiddlers

Baddies:

- The fiddlers plays for old King Cole
- He slept the fiddlers
- Old King Cole called for he

Context free assumption



$\boxed{\text{Verb} \rightarrow \text{was} \mid \text{called} \mid \text{plays} \mid \text{play} \mid \text{slept}}$ how do we fix this?

Updating the lexicon

SgNoun	$\longrightarrow$	soul $\mid$ pipe $\mid$ bowl
PlNoun	$\longrightarrow$	fiddlers
SgProperNoun	$\longrightarrow$	King Cole
3rdSgPronoun	$\longrightarrow$	he
3rdPlPronoun	$\longrightarrow$	they
3rdPlPronoun	$\longrightarrow$	I
SgArticle	$\longrightarrow$	a $\mid$ the
PlArticle	$\longrightarrow$	the

1stSgIntrans	$\longrightarrow$	sleep
3rdSgIntrans	$\longrightarrow$	sleeps
3rdPlIntrans	$\longrightarrow$	sleep
1stSgTrans	$\longrightarrow$	play
3rdSgTrans	$\longrightarrow$	plays
3rdPlTrans	$\longrightarrow$	play

Updating the syntactic rules

Original:

NP	$\rightarrow$	Adjective ProperNoun
	|	Possessive Adjective Noun
	|	Article Adjective* Noun
	|	Pronoun

Updated:

3rdSgNP	$\rightarrow$	Adjective SgProperNoun
	|	Possessive Adjective SgNoun
	|	SgArticle Adjective* SgNoun
	|	3rdSgPronoun

elegant?

Unification grammar

$$\text{Pronoun} \rightarrow \text{I}$$
$$\begin{array}{lll} \text{num} & = & \text{SG} \\ \text{person} & = & 1 \end{array}$$

$$\left[\begin{array}{ll} \textbf{I (type)} & \\ \text{num}: & \text{sg} \\ \text{person}: & 1 \end{array}\right]$$

CFG rules are augmented with constraints and

their instances are augmented with feature structures.

Complex feature structures

NP $\longrightarrow$ Article Noun

self.agreement	=	Noun.agreement
Noun.agreement	=	Article.agreement

$$
\begin{bmatrix}
\textbf{NP} & \\
\text{agreement :} & \boxed{1} \begin{bmatrix} \textbf{Agreement} \\ \text{num : sg} \\ \text{person : 1} \end{bmatrix} \\
\text{Article :} & \begin{bmatrix} \text{agreement : } \boxed{1} \end{bmatrix} \\
\text{Noun :} & \begin{bmatrix} \text{agreement : } \boxed{1} \end{bmatrix}
\end{bmatrix}
$$

unification/coindexation

Feature structure unification

A simple recursive algorithm that checks to see if two feature structures are compatible.

- Base case: if two atomic values are the same, then they unify
- Recursive case: if two feature structures have features that unify, then the two feature structures unify.
- Otherwise, return failure

Unification success

$$\begin{bmatrix} \text{agreement}: & \boxed{1}\begin{bmatrix}\text{person}: & 3\end{bmatrix} \\ \text{subject}: & \begin{bmatrix}\text{case}: & \text{nom}\\ \text{agreement}: & \boxed{1}\end{bmatrix}\end{bmatrix}$$

unified with

$$\begin{bmatrix}\text{subject}: & \begin{bmatrix}\text{case}: & \\ \text{agreement}: & \begin{bmatrix}\text{person}: & 3\\ \text{number}: & \text{SG}\end{bmatrix}\end{bmatrix}\end{bmatrix} =$$

$$\begin{bmatrix} \text{agreement}: & \boxed{1} \\ \text{subject}: & \begin{bmatrix}\text{case}: & \text{nom}\\ \text{agreement}: & \boxed{1}\begin{bmatrix}\text{person}: & 3\\ \text{number}: & \text{SG}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Unification failure

$$\begin{bmatrix} \text{agreement} : & \boxed{1} \\ \text{subject} : & \begin{bmatrix} \text{case} : & \text{nom} \\ \text{agreement} : & \boxed{1} \end{bmatrix} \end{bmatrix} \quad \text{unified with}$$

$$\begin{bmatrix} \text{subject} : & \begin{bmatrix} \text{case} : & \\ \text{agreement} : & \begin{bmatrix} \text{person} : & 3 \\ \text{number} : & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix} =$$

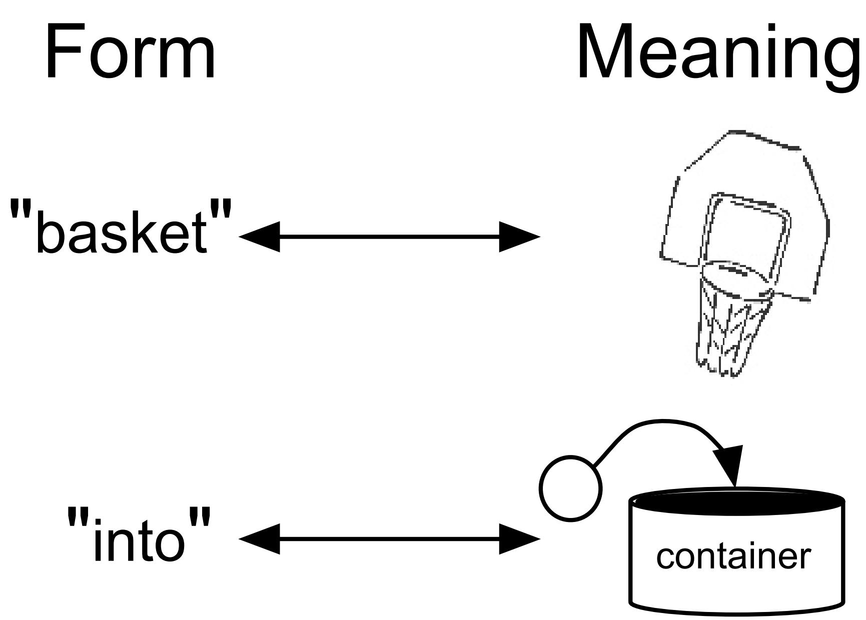
FAILURE

What are we missing?

Still no semantics!

Construction grammar

- Grammaticality = form *and* function
- Each rule (construction) is a set of form/meaning constraints (a pair)
- Not purely compositional
- Implies early use of semantics in processing



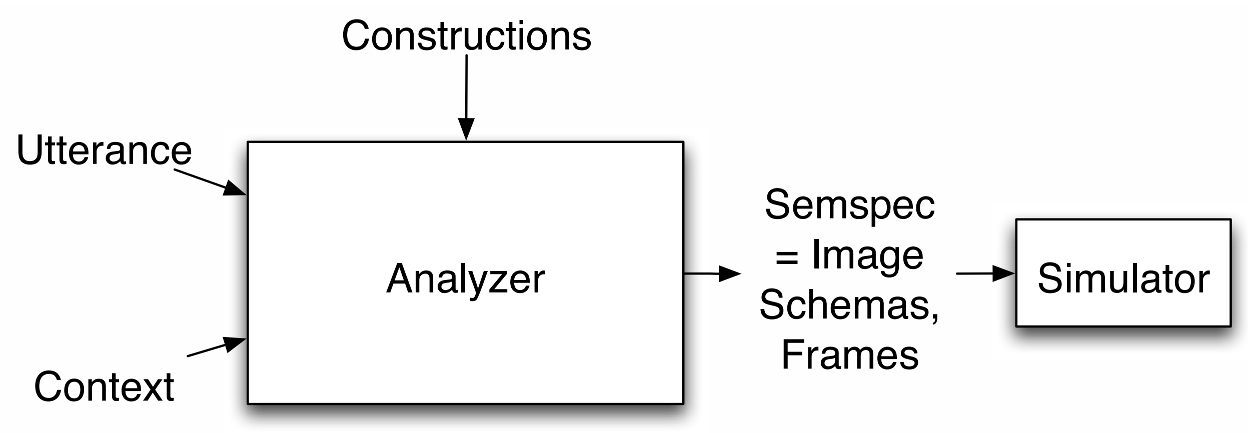
Example Constructions

- "-ed" $\leftrightarrow$ past speech time, completed action
- "give" $\leftrightarrow$ a give action
- $NP_0\ V\ NP_1\ NP_2$ $\leftrightarrow$ Transfer Scene + bindings (Goldberg)
- WXDY $\leftrightarrow$ How come X is doing Y? (Kay, Fillmore)
- The *There* Constructions (Lakoff)

Each has a form pole and a meaning pole.

ECG overview

- Embodied + Construction Grammar
- Precise, unification based (LFG, HPSG, CxG)
- Simulation
- Primitives: schemas and constructions
- Schemas and constructions form inheritance hierarchies.



Embodied schemas in ECG

schema TL
trajector
landmark

schema SPG
subcase of TL
source
path
goal

schema Container
interior
exterior
boundary
portal

Actions

```
schema ForceApplicationAction
  subcase of Action
  evokes ForceTransfer as forceTransfer
  roles
    actor (inherited)
    actedUpon
  constraints
    actor ⟷ forceTransfer.supplier
    actedUpon ⟷ forceTransfer.recipient
```

ECG's KR tools: subcase of, roles, evokes, constraints

An *Into* construction

construction Into
  subcase of DynamicSpatialPrep
  form : “into”
  meaning : SPG
    evokes Container **as** c
    self$_m$.goal $\longleftrightarrow$ c.interior
    self$_m$.source $\longleftrightarrow$ c.exterior
    self$_m$.landmark $\longleftrightarrow$ c

An *ActiveSelfMotion* construction

construction ActiveSelfMotion
  subcase of ActiveVP
  constructional
    constituents
      v **:** Verb
      pp **:** SpatialPP
  form
    v_f `before` pp_f
  meaning : SelfMotionPathEvent
    self$_m$.profiled-participant $\longleftrightarrow$ **self**$_m$.mover
    self$_m$.profiled-process $\longleftrightarrow$ **self**$_m$.la
    self$_m$.profiled-process $\longleftrightarrow$ v_m
    self$_m$.spg $\longleftrightarrow$ pp_m

What's neural about all this?

Nothing inherently, but if you use it right, plenty:

- Embodied schemas with neurally plausible reductions
- Feature structures and bindings are neurally reducible
- Simulation/X-schemas
- Probabilistic ECG