Meanings First

Context and Content Lectures, Institut Jean Nicod

June 6: General Introduction and "Framing Event Variables"

June 13: "I-Languages, T-Sentences, and Liars"

June 20: "Words, Concepts, and Conjoinability"
[about 1/3 of the posted slides, but a lot of the content]

June 27: "Meanings as Concept Assembly Instructions"

Main Idea: Short Form

- In acquiring words, kids use available concepts to *introduce* new ones.

  Sound('ride') + RIDE(_, _) ==> RIDE(_) + RIDE(_, _) + 'ride'

- Meanings are *instructions* for how to access and combine *i-concepts*

  -- lexicalizing RIDE(_, _) puts RIDE(_) at an accessible address

  -- introduced concepts can be constituents of (variable-free) *conjunctions* that are formed without a Tarskian ampersand

'fast horses' FAST( )^HORSES( ) ←→ FAST( )^HORSE( )^PLURAL(_)

'ride horses' RIDE( )^∃[Θ(, _)^HORSES(_)]

Lots of Conjoiners

- $P \,\&\, Q$ — purely propositional
- $Fx \,\&^{M}\, Gx$ — purely monadic
- ??? — ???
- $Rx_1x_2 \,\&^{DF}\, Sx_1x_2$ — purely dyadic, with fixed order

  $Rx_1x_2 \,\&^{DA}\, Sx_2x_1$ — purely dyadic, any order
- $Rx_1x_2 \,\&^{PF}\, Tx_1x_2x_3x_4$ — polyadic, with fixed order

  $Rx_1x_2 \,\&^{PA}\, Tx_3x_4x_1x_5$ — polyadic, any order

  $Rx_1x_2 \,\&^{PA}\, Tx_3x_4x_5x_6$ — *the number of variables in the conjunction can exceed the number in either conjunct*

NOT EXTENSIONALLY EQUIVALENT

Lots of Conjoiners

- P & Q — purely propositional
- Fx $\&^{M}$ Gx — purely monadic

  Fx^Gx ; Rex^Gx — G(_) can “join” with F(_) or R(, _)

- Rx_1x_2 $\&^{DF}$ Sx_1x_2 — purely dyadic, with fixed order

  Rx_1x_2 $\&^{DA}$ Sx_2x_1 — purely dyadic, any order

- Rx_1x_2 $\&^{PF}$ $Tx_1x_2x_3x_4$ — polyadic, with fixed order

  Rx_1x_2 $\&^{PA}$ $Tx_3x_4x_1x_5$ — polyadic, any order

  Rx_1x_2 $\&^{PA}$ $Tx_3x_4x_5x_6$ — *the number of variables in the conjunction can exceed the number in either conjunct*

Main Idea: Short Form

- In acquiring words, kids use available concepts to *introduce* new ones.

  Sound('ride') + RIDE(_, _) ==> RIDE(_) + RIDE(_, _) + 'ride'

- Meanings are *instructions* for how to access and combine *i-concepts*

  -- lexicalizing RIDE(_, _) puts RIDE(_) at an accessible address

  -- introduced concepts can be constituents of (variable-free) *conjunctions* that are formed without a Tarskian ampersand

'fast horses' FAST( )^HORSES( )

'ride horses' RIDE( )^∃[Θ(, _)^HORSES(_)]

'her ride horses' ∃[Θ2(, _)^HER(_)]^RIDE( )^∃[Θ(, _)^HORSES(_)]

ext *int*

Main Idea: Short Form

- In acquiring words, kids use available concepts to ***introduce*** new ones.

  Sound('ride') + RIDE(_, _) ==> RIDE(_) + RIDE(_, _) + 'ride'

- Meanings are ***instructions*** for how to access and combine ***i-concepts***

  -- lexicalizing RIDE(_, _) puts RIDE(_) at an accessible address

  -- introduced concepts can be constituents of (variable-free) *conjunctions* that are formed without a Tarskian ampersand

But what about...

*Chris devoured

*Chris put the book

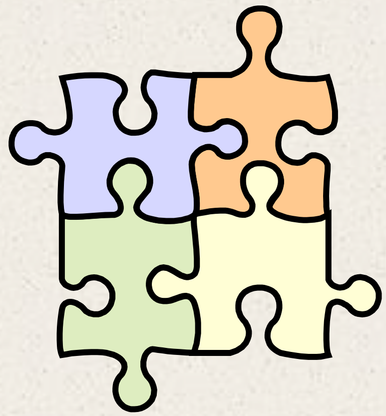
*Brutus sneezed Caesar

*Brutus arrived Caesar (to) Antony

Conceptual Adicity

Two Common Metaphors

- Jigsaw Puzzles



- 7th Grade Chemistry $^{+1}H - \overset{-2}{O} - H^{+1}$

Jigsaw Metaphor

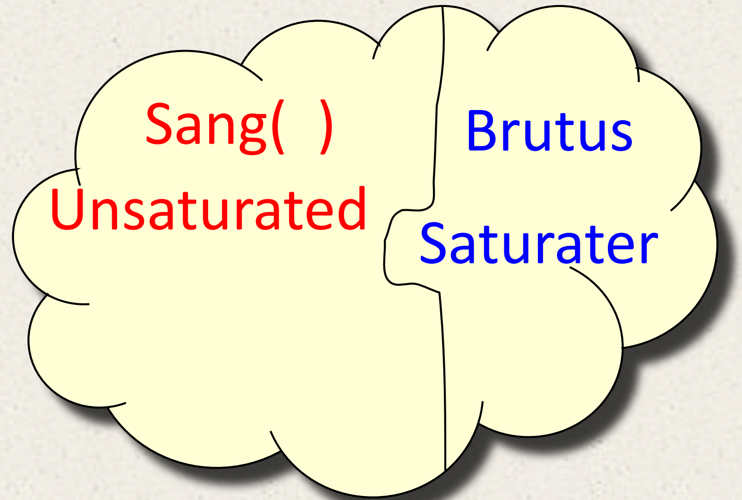
A
THOUGHT

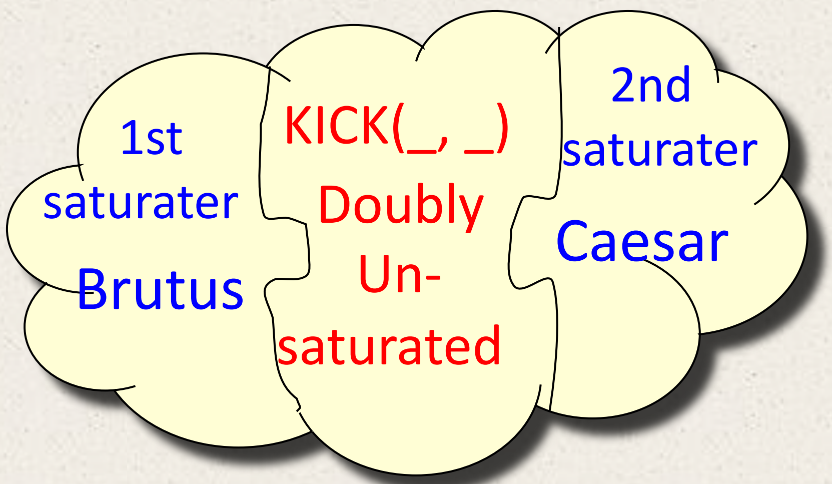
Jigsaw Metaphor

one Dyadic Concept
(adicity: -2)

“filled by” two Saturaters
(adicity +1)

yields a complete Thought





one Monadic Concept
(adicity: -1)

“filled by” one Saturater
(adicity +1)

yields a complete Thought

7^{th} Grade Chemistry Metaphor

-2

$^{+1}H(OH^{+1})^{-1}$

a single atom with valence -2
can combine with
two atoms of valence +1
to form a stable molecule

7th Grade Chemistry Metaphor

-2

$^{+1}$Brutus(KickCaesar^{+1})$^{-1}$

7th Grade Chemistry Metaphor

$^{+1}$BrutusSang^{-1}

$^{+1}NaCl^{-1}$

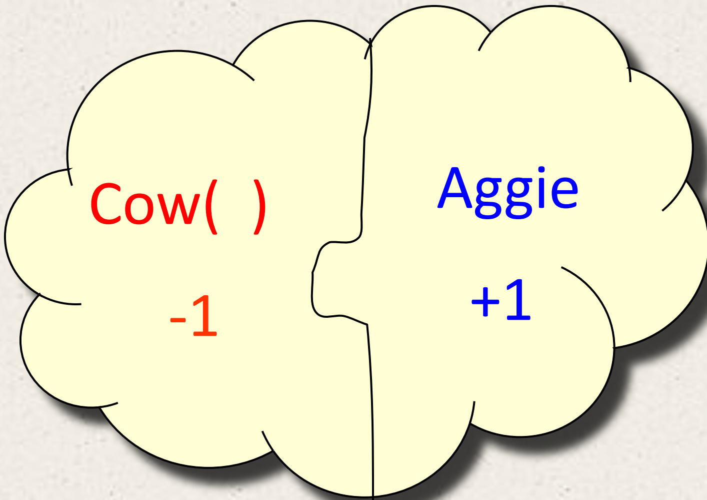
an atom with valence -1

can combine with

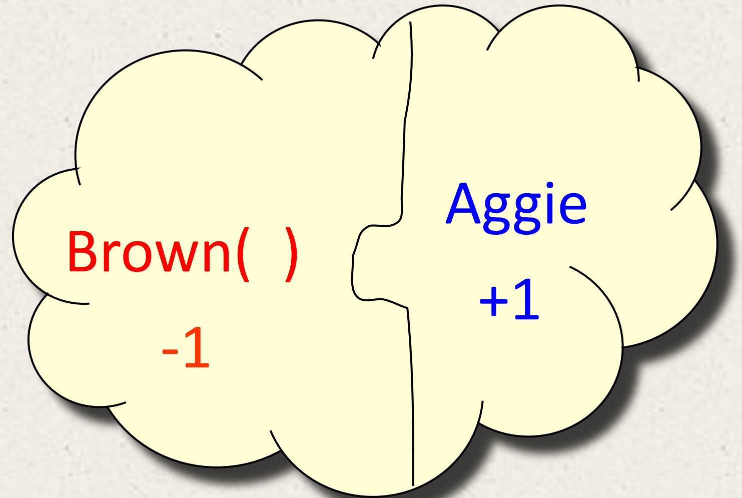
an atom of valence +1

to form a stable molecule

Extending the Metaphor

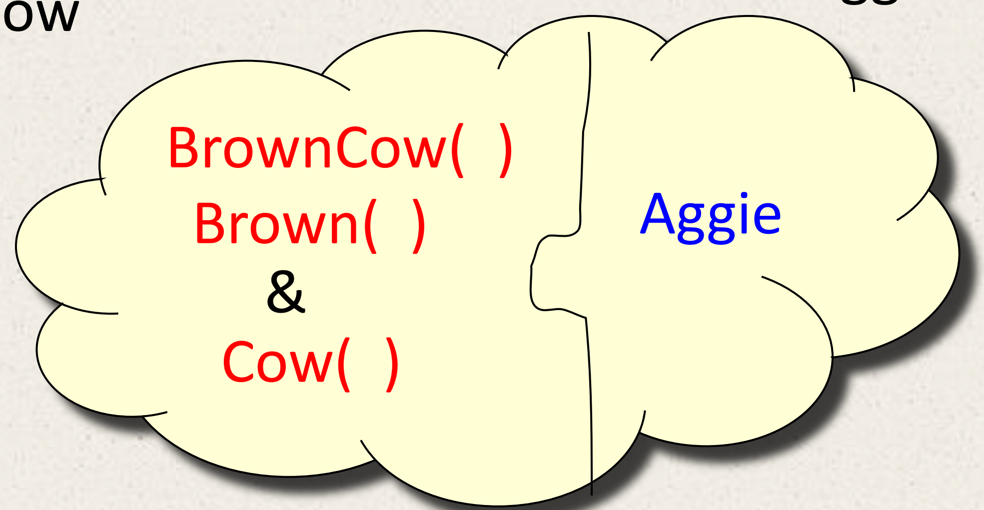


Aggie is (a) cow

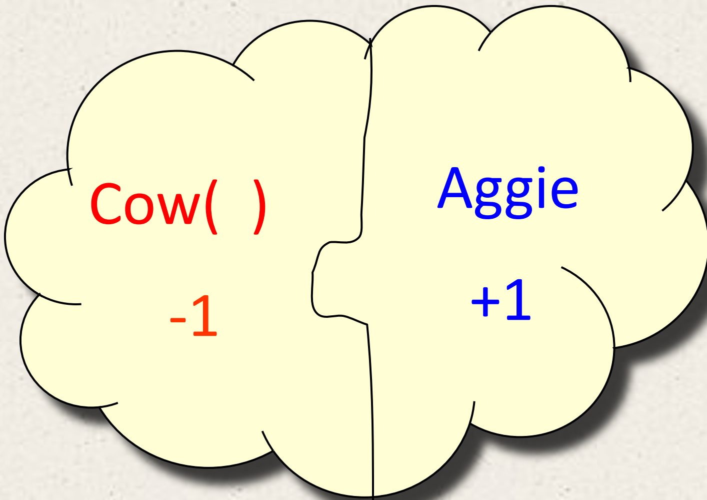


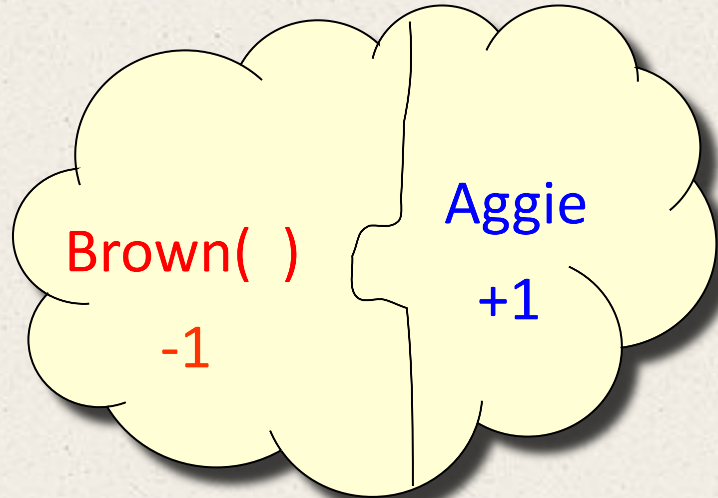
Aggie is brown

Aggie is (a) brown cow

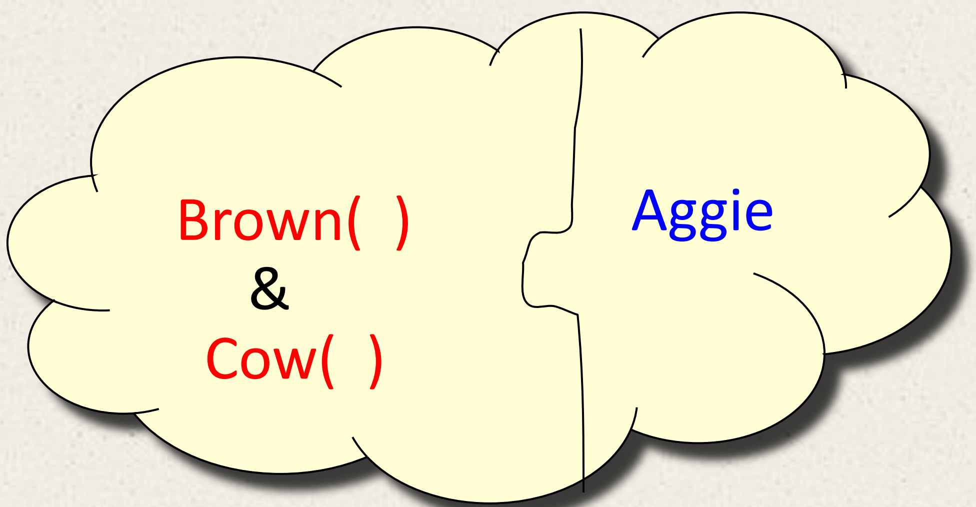


Extending the Metaphor





Conjoining two monadic (-1) concepts can yield a complex monadic (-1) concept



Conceptual Adicity

TWO COMMON METAPHORS

--Jigsaw Puzzles

--7th Grade Chemistry

DISTINGUISH

Lexicalized concepts, L-concepts

RIDE(_, _)   GIVE(_, _, _)   ALVIN

Introduced concepts, I-concepts

RIDE(_)   GIVE(_)   CALLED(_, *Sound*('Alvin'))

my hypothesis: I-concepts exhibit *less typology* than L-concepts

special case: I-concepts exhibit *fewer adicities* than L-concepts

A Different (older) Hypothesis

Words *Label* Concepts

Sound('ride') + RIDE(_, _) ==> RIDE(_, _) + 'ride'

Sound('Alvin') + ALVIN ==> ALVIN + 'Alvin'

- Acquiring words is basically a process of pairing perceptible signals with *pre-existing* concepts
- Lexicalization is a conceptually *passive* operation
- Word combination mirrors concept combination

Bloom: *How Children Learn the Meanings of Words*

- word meanings are, at least primarily,
  concepts that kids have *prior* to lexicalization

- learning word meanings is, at least primarily,
  a process of figuring out *which* concepts
  are paired with *which* word-sized signals

- in this process, kids draw on many capacities—e.g.,
  recognition of *syntactic cues* and *speaker intentions*—
  but no capacities *specific to acquiring word meanings*

Lidz, Gleitman, and Gleitman

"Clearly, the number of noun phrases required for the grammaticality of a verb in a sentence is a function of the number of participants logically implied by the verb meaning.

It takes only one to sneeze, and therefore *sneeze* is intransitive, but it takes two for a kicking act (kicker and kickee), and hence *kick* is transitive.

Of course there are quirks and provisos to these systematic form-to-meaning-correspondences…"

Lidz, Gleitman, and Gleitman

"Clearly, the number of noun phrases required for the grammaticality of a verb in a sentence is a function of the number of participants logically implied by the verb meaning.

It takes only one to sneeze, and therefore *sneeze* is intransitive, but it takes two for a kicking act (kicker and kickee), and hence *kick* is transitive.

Of course there are quirks and provisos to these systematic form-to-meaning-correspondences…"

Another Perpsective...

Clearly, the number of noun phrases required for the grammaticality of a verb in a sentence is not a function of the number of participants logically implied by the verb meaning.

A *paradigmatic* act of kicking has exactly two participants (kicker and kickee), and yet *kick* need not be transitive.

Brutus kicked Caesar the ball

Caesar was kicked

Brutus kicked

Brutus gave Caesar a swift kick

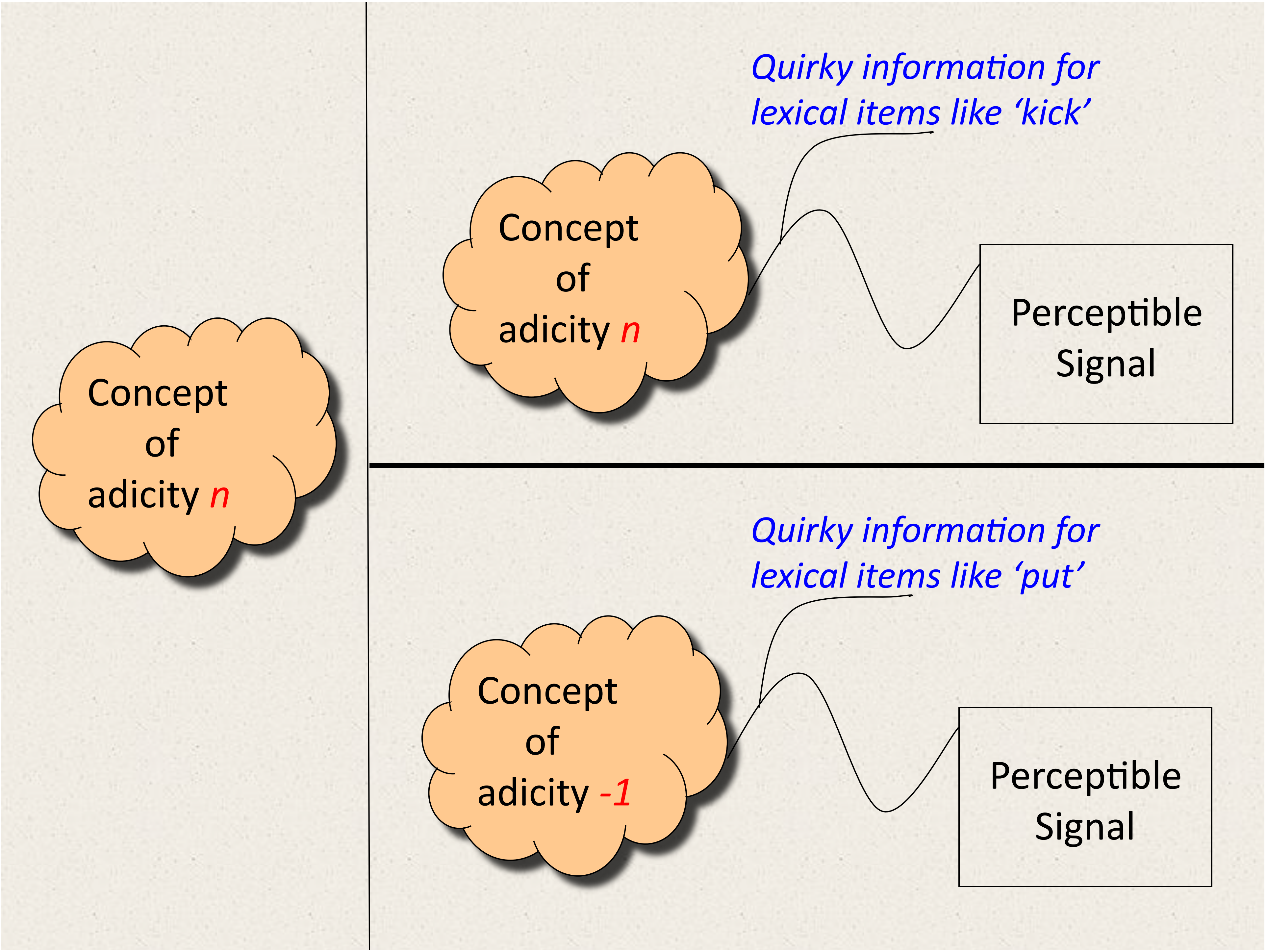
*Brutus put the ball

*Brutus put

*Brutus sneezed Caesar

*Brutus devoured

Of course there are quirks and provisos. Some verbs do require a certain number of noun phrases in active voice sentences.


Concept of adicity n
Quirky information for lexical items like ‘kick’
Concept of adicity n
Perceptible Signal
Quirky information for lexical items like ‘put’
Concept of adicity -1
Perceptible Signal

Clearly, the number of noun phrases required for the grammaticality of a verb in a sentence is a function of the number of participants logically implied by the verb meaning.

It takes only one to sneeze, and therefore *sneeze* is intransitive, but it takes two for a kicking act (kicker and kickee), and hence *kick* is transitive.

Of course there are quirks and provisos to these systematic form-to-meaning-correspondences.

Clearly, the number of noun phrases required for the grammaticality of a verb in a sentence isn't a function of the number of participants logically implied by the verb meaning.

It takes only one to sneeze, and usually *sneeze* is intransitive. But it usually takes two to have a kicking; and yet *kick* can be untransitive.

Of course there are quirks and provisos. Some verbs do require a certain number of noun phrases in active voice sentences.

Clearly, the number of noun phrases required for the grammaticality of a verb in a sentence is a function of the number of participants logically implied by the verb meaning.

It takes only one to sneeze, and therefore *sneeze* is intransitive, but it takes two for a kicking act (kicker and kickee), and hence *kick* is transitive.

Of course there are quirks and provisos to these systematic form-to-meaning-correspondences.

Clearly, the number of noun phrases required for the grammaticality of a verb in a sentence isn't a function of the number of participants logically implied by the verb meaning.

It takes only one to sneeze, and *sneeze* is typically used intransitively; but a paradigmatic kicking has exactly two participants, and yet *kick* can be used intransitively or ditransitively.

Of course there are quirks and provisos. Some verbs do require a certain number of noun phrases in active voice sentences.

Quirks and Provisos, or Normal Cases?

KICK(x_1, x_2)	The baby kicked
RIDE(x_1, x_2)	Can you give me a ride?
BEWTEEN(x_1, x_2, x_3)	I *am* between him *and* her
	why not: I between him her
BIGGER(x_1, x_2)	This *is* bigg*er* *than* that
	why not: This bigs that
MORTAL(…?...)	Socrates is mortal
	A mortal wound is fatal
FATHER(…?...)	Fathers father
	Fathers father future fathers
EAT/DINE/GRAZE(…?...)	

OK, but what about…

(1) *Chris devoured

(2) *Chris put the book

(3) *Brutus sneezed Caesar

(4) *Brutus arrived Caesar (to) Antony

OK, but what about…

(1) *Chris devoured

(2) *Chris put the book

(1a) Chris devoured the pizza

(1b) Chris ate

(1c) Chris ate the pizza

if (1) is *un*acceptable because ‘devoured’ lexicalized DEVOURED(x, y) and so this verb has valence -2, then why are (1b) *and* (1c) acceptable?

if (2) is *un*acceptable because ‘put’ lexicalized PUT(x, y, z) and so this verb has valence of -3, then a verb whose valence is –*n* can take *fewer* than *n* grammatical *arguments*

OK, but what about…

(1) *Chris devoured

(2) *Chris put the book

(1a) Chris devoured the pizza

(1b) Chris ate

(1c) Chris ate the pizza

if (1) and (2) are *un*acceptable because verbal valences are unsatisfied,
then a "single" verb ('ate', 'kick', ...) can have different "valence forms,"
and valence requirements can sometimes be satisfied by adjuncts

Another way of encoding the constrasts

'devoured' fetches a monadic concept; but it *also* imposes a [+Patient] requirement on phrases, partly because it lexicalized a certain dyadic concept

OK, but what about…

(1) *Chris devoured

(2) *Chris put the book

(1a) Chris devoured the pizza

(1b) Chris ate

(1c) Chris ate the pizza

if (1) and (2) are ***un***acceptable because verbal valences are unsatisfied, then a "single" verb ('ate', 'kick', ...) can have different "valence forms," and valence requirements can sometimes be satisfied by adjuncts

Another way of encoding the constrasts

'put' fetches a monadic concept; but it ***also*** imposes a [+Patient, +Loc] requirement on phrases, partly because it lexicalized a certain dyadic concept

OK, but what about…

(1) *Chris devoured

(2) *Chris put the book

(1a) Chris devoured the pizza

(1b) Chris ate

(1c) Chris ate the pizza

Sometimes, unacceptability is just idiosyncracy

*Chris goed to the store

(1d) Chris dined

(1e) *Chris dined the pizza

(1f) Chris dined on shrimp

(1g) *Chris devoured on shrimp

(2a) ? Chris placed the book

(2b) Chris placed the book nicely

OK, but what about…

(1) *Chris devoured

(2) *Chris put the book

(1a) Chris devoured the pizza

(1b) Chris ate

(1c) Chris ate the pizza

if (1) and (2) are ***un***acceptable because verbal valences are unsatisfied,
then a "single" verb ('ate', 'kick', ...) can have different "valence forms,"
and valence requirements can sometimes be satisfied by adjuncts

Don't encode idiosyncracies as structural requirements.
This makes a mystery of flexibility ***and*** idiosyncracy.
Distinguish ***structural requirements*** from ***filters***.

A verb can access a monadic concept *and*
impose further (idiosyncratic) restrictions on complex expressions

- *Semantic Composition Adicity Number* (SCAN)

(instructions to fetch) singular concepts	+1	<e>
(instructions to fetch) monadic concepts	-1	<e, t>
(instructions to fetch) dyadic concepts	-2	<e,<e, t>>

- *Property of Smallest Sentential Entourage* (POSSE)

  zero NPs, one NP, two NPs, …

the SCAN of every verb can be -1, while POSSEs vary: zero, one, two, …

a verb's POSSE may reflect

...the adicity of the concept *lexicalized*

…whether or not this concept is itself "thematically rich"

...*statistics* about how verbs are *used* (e.g., in active voice)

...*prototypicality* effects

...other *agrammatical* factors

- 'put' may have a (lexically represented) POSSE of three *in part because*

  --the concept lexicalized was PUT(_, _, _)

  --this concept is relatively "bleached"

  --the frequency of locatives (as in 'put the cup on the table') is salient

On any view: Two Kinds of Facts to Accommodate

- *Flexibilities*
  - Brutus kicked Caesar
  - Caesar was kicked
  - The baby kicked
  - I get a kick out of you
  - Brutus kicked Caesar the ball
- *Inflexibilities*
  - Brutus put the ball on the table
  - *Brutus put the ball
  - *Brutus put on the table

On any view: Two Kinds of Facts to Accommodate

- *Flexibilities*
  - The coin melted
  - The jeweler melted the coin
  - The fire melted the coin
  - The coin vanished
  - The magician vanished the coin
- *Inflexibilities*
  - Brutus arrived
  - *Brutus arrived Caesar

OK, but what about…

(3) *Brutus sneezed Caesar

(4) *Brutus arrived Caesar (to) Antony

Well…

Brutus burped Caesar

Brutus vanished Caesar

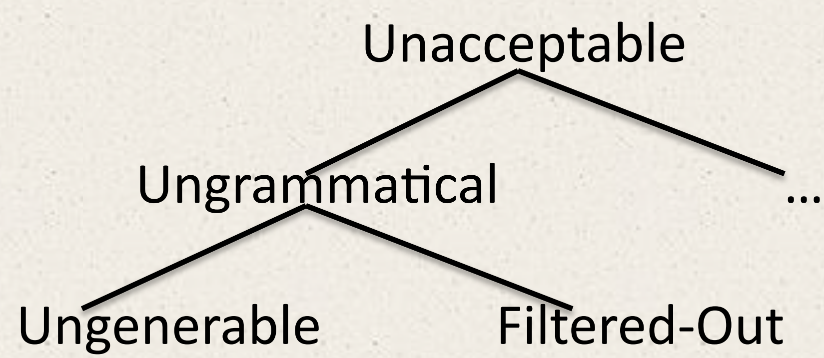
Brutus sent Caesar Antony

Brutus sent for help

*Brutus goed to the store

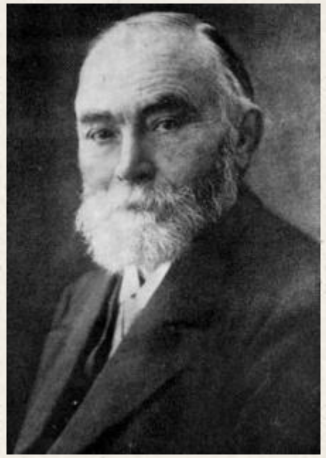
*Brutus seems sleeping

*Brutus kicked that Caesar arrived



Lexicalization as Concept-Introduction (not mere labeling)

Concept of type T



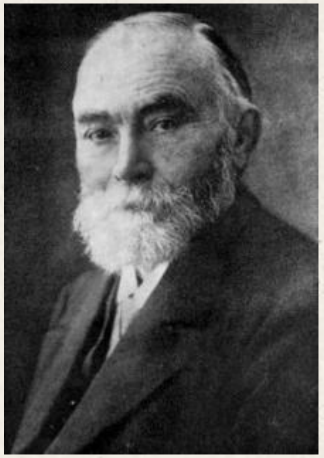
Concept of type T

Concept of type T*

Perceptible Signal

Lexicalization as Concept-Introduction (not mere labeling)

Number(_)
type: <e, **t**>

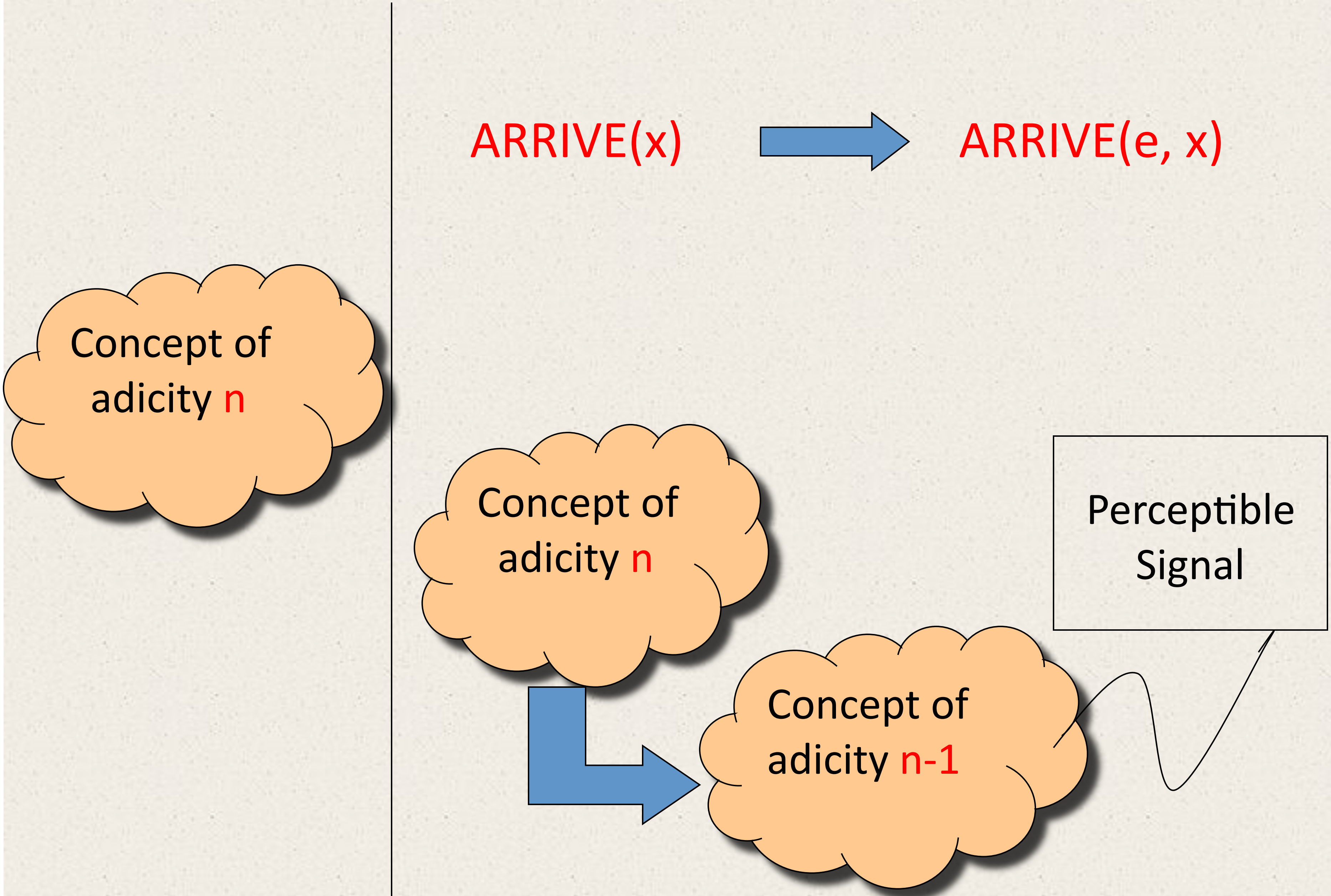


Number(_)
type: <e, **t**>

Perceptible Signal

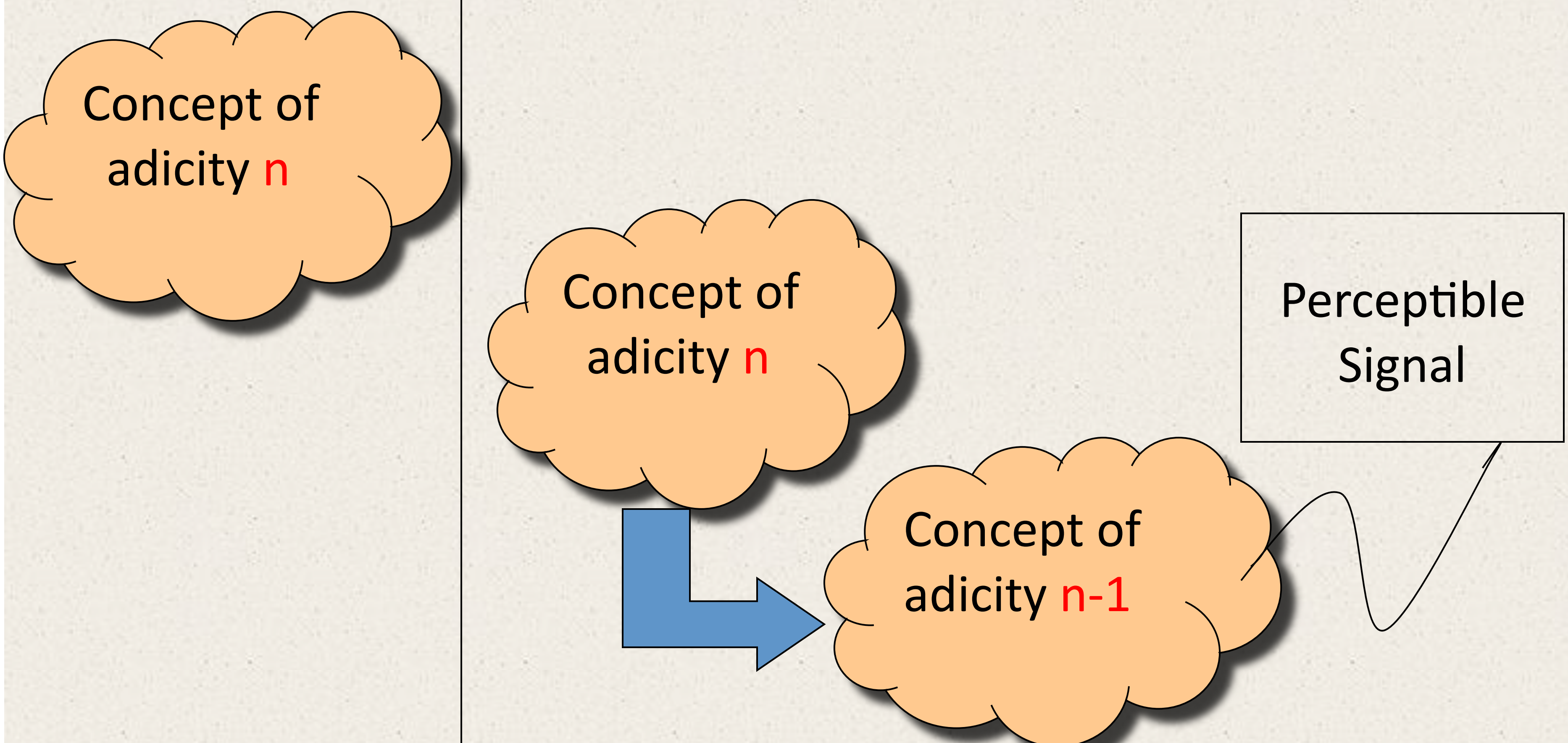
NumberOf[_, Φ(_)]
type: <<e, **t**>, <*n*, **t**>>

One Possible (Davidsonian) Application: Increase Adicity

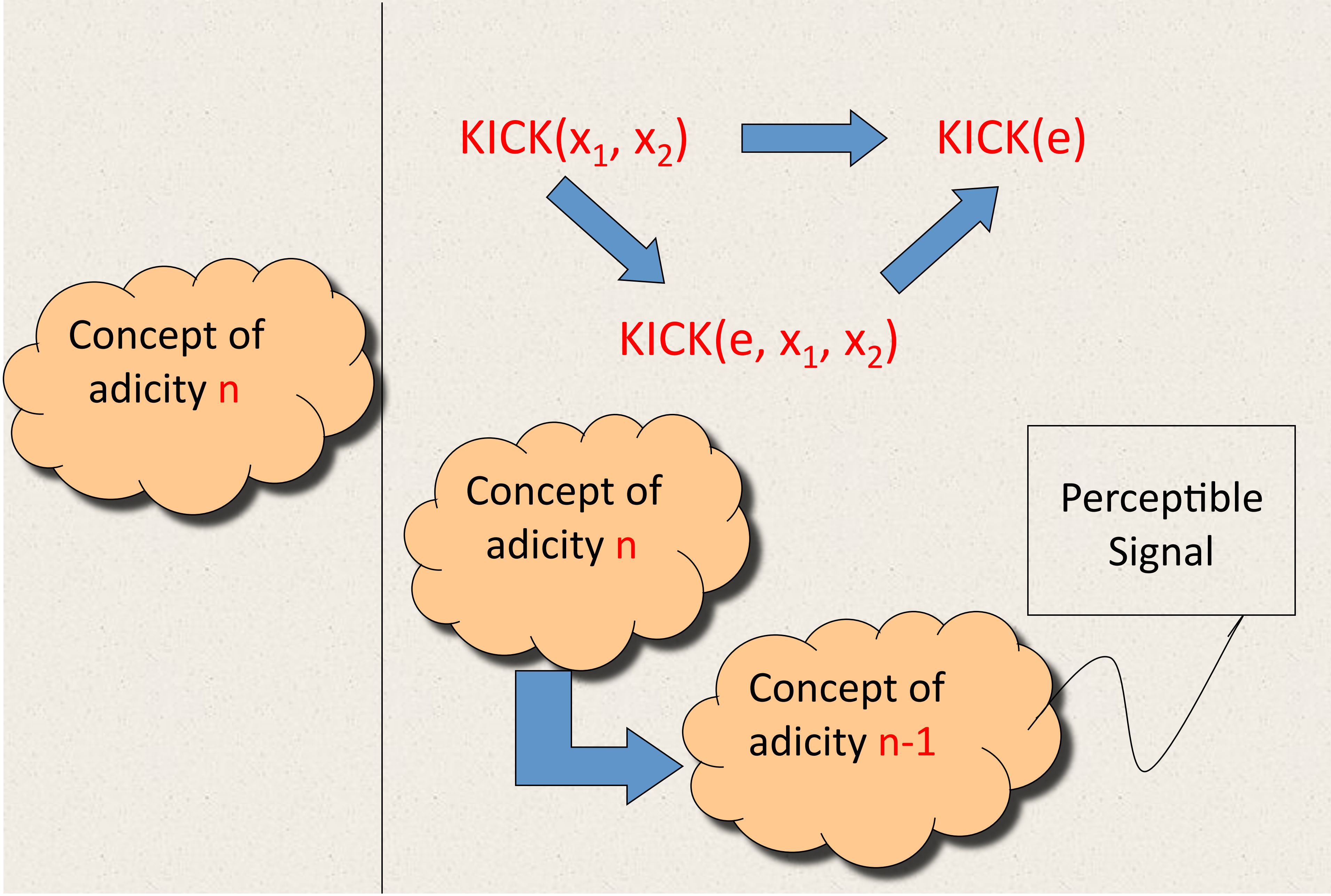


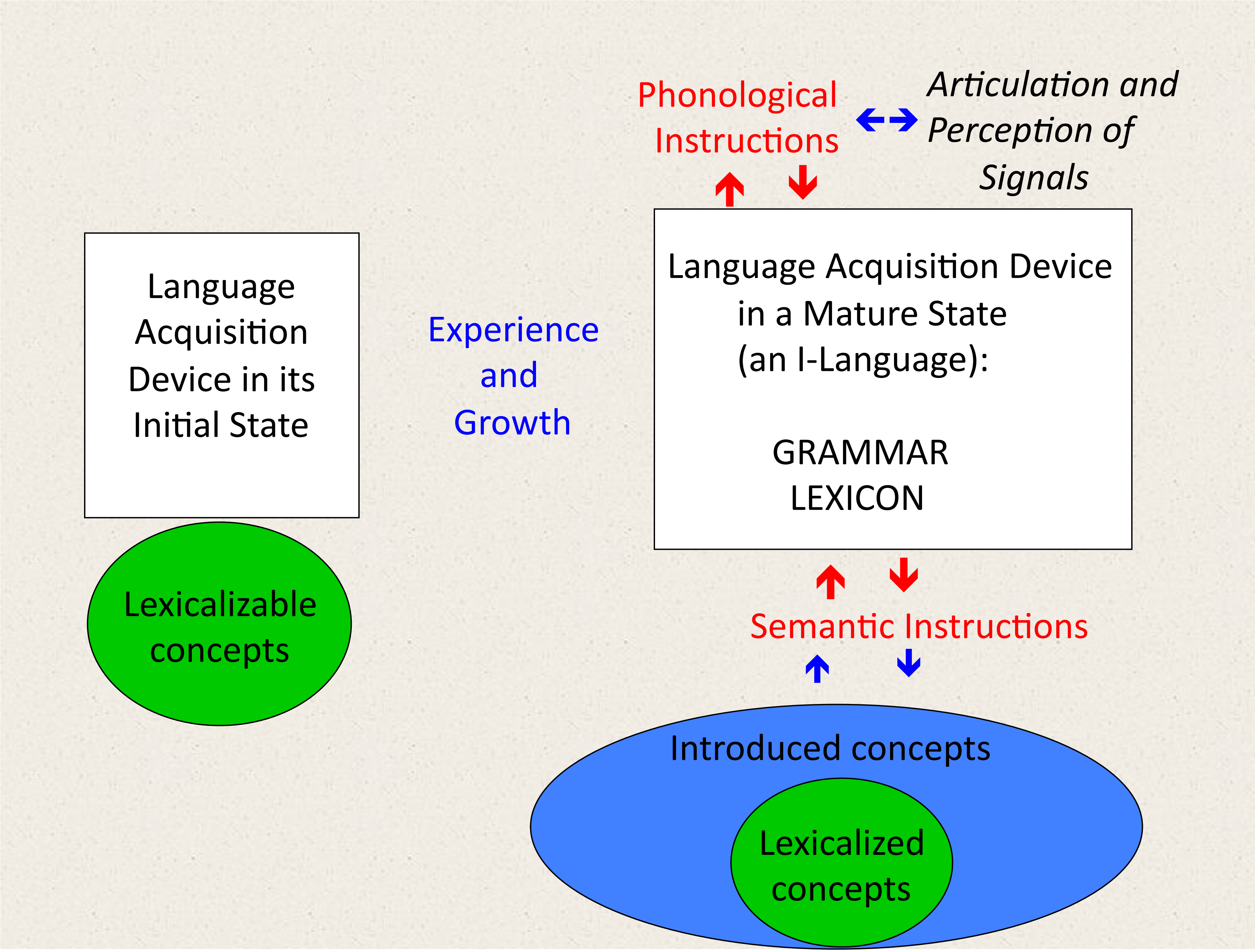
One Possible (Davidsonian) Application: Increase Adicity

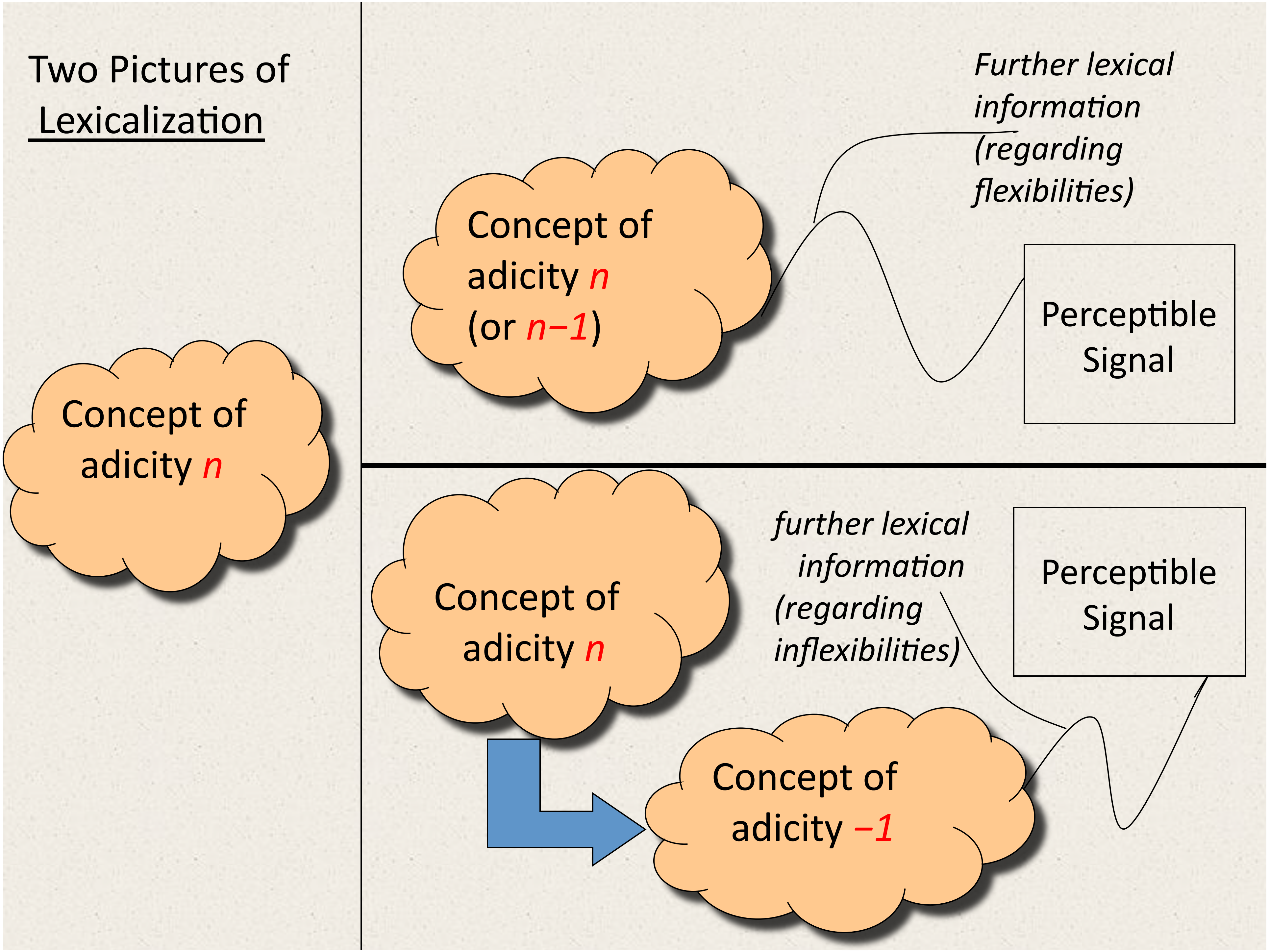
$\mathrm{KICK}(x_1, x_2) \rightarrow \mathrm{KICK}(e, x_1, x_2)$



Another Possible Application: Make Monads

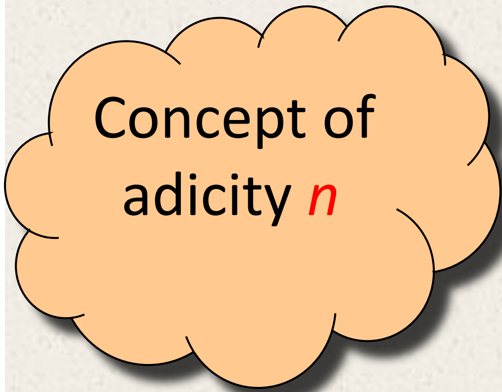


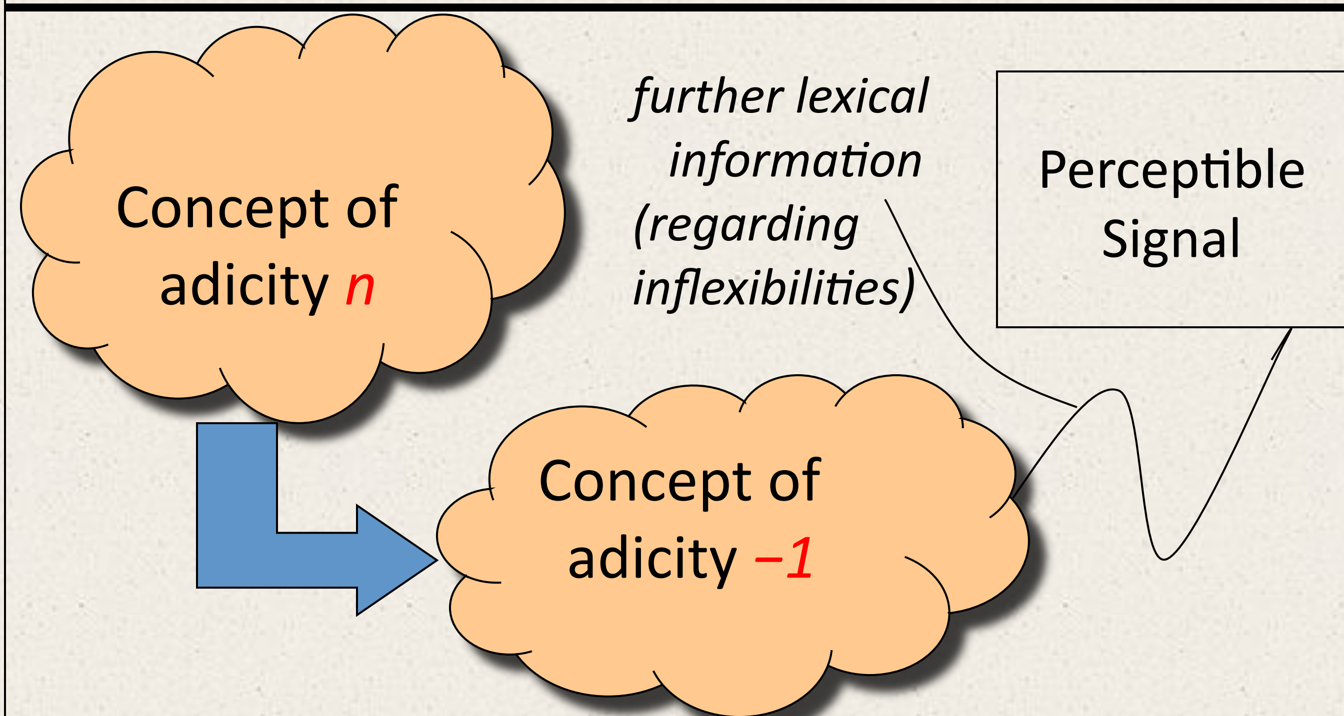

Phonological Instructions
Articulation and Perception of Signals
Language Acquisition Device in its Initial State
Experience and Growth
Language Acquisition Device in a Mature State (an I-Language):
GRAMMAR
LEXICON
Lexicalizable concepts
Semantic Instructions
Introduced concepts
Lexicalized concepts


Two Pictures of
Lexicalization
Concept of
adicity *n*
Concept of
adicity *n*
(or *n*−*1*)
Further lexical
information
(regarding
flexibilities)
Perceptible
Signal
Concept of
adicity *n*
further lexical
information
(regarding
inflexibilities)
Perceptible
Signal
Concept of
adicity −*1*

Two Pictures of Lexicalization

offer some reminders of the reasons for adopting the second picture





Absent Word Meanings

Striking *absence* of certain (open-class) lexical meanings
that *would* be permitted
if Human I-Languages permitted *non*monadic semantic types

<e,<e,<e,<e, **t**>>>> (instructions to fetch) tetradic concepts

<e,<e,<e, **t**>>> (instructions to fetch) triadic concepts

<e,<e, **t**>> (instructions to fetch) dyadic concepts

<e> (instructions to fetch) singular concepts

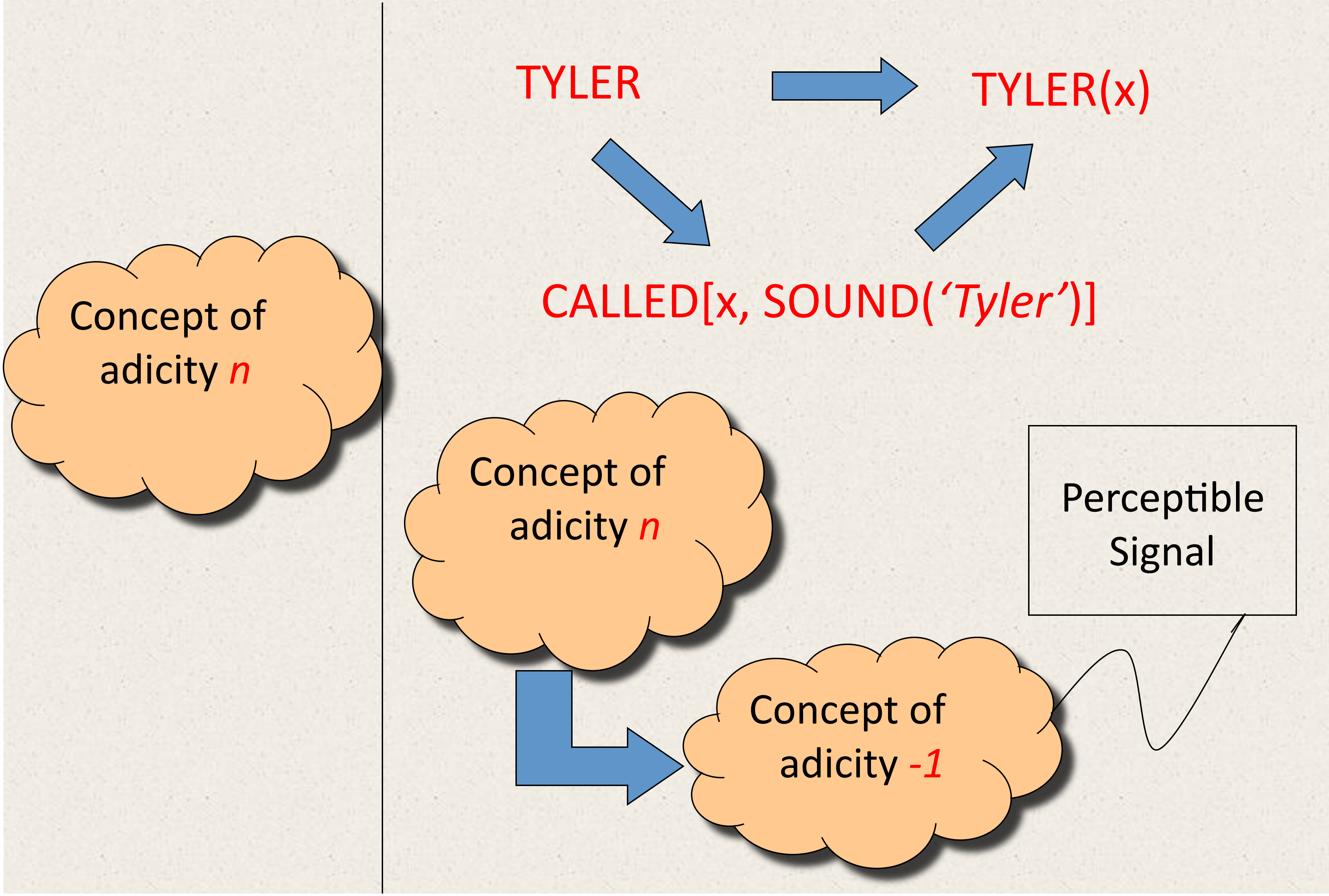
<u>Proper Nouns</u>

- even English tells against the idea that <u>*lexical proper nouns*</u> label singular concepts (of type <e>)

- Every Tyler I saw was a philosopher
  Every philosopher I saw was a Tyler
  There were three Tylers at the party
  That Tyler stayed late, and so did this one
  Philosophers have wheels, and Tylers have stripes
  The Tylers are coming to dinner
  I spotted Tyler Burge
  I spotted that nice Professor Burge who we met before

- proper nouns seem to fetch monadic concepts,
  even if they lexicalize singular concepts

Lexicalization as Concept-Introduction: Make Monads



Absent Word Meanings

Striking *absence* of certain (open-class) lexical meanings

that *would* be permitted

if I-Languages permit nonmonadic semantic types

<e,<e,<e,<e, **t**>>>> (instructions to fetch) tetradic concepts

<e,<e,<e, **t**>>> (instructions to fetch) triadic concepts

<e,<e, **t**>> (instructions to fetch) dyadic concepts

<e> (instructions to fetch) singular concepts

Absent Word Meanings

Brutus sald a car Caesar a dollar

sald	➔ SOLD(x, $, z, y)	x sold y to z (in exchange) for $
[sald [a car]]	➔ SOLD(x, $, z, a car)	
[[sald [a car]] Caesar]	➔ SOLD(x, $, Caesar, a car)	
[[[sald [a car]] Caesar]] a dollar]	➔ SOLD(x, a dollar, Caesar, a car)	

Caesar bought a car

bought a car from Brutus for a dollar

bought Antony a car from Brutus for a dollar

Absent Word Meanings

Brutus tweens Caesar Antony

tweens	→ BETWEEN(x, z, y)
[tweens Caesar]	→ BETWEEN(x, z, Caesar)
[[tweens Caesar] Antony]	→ BETWEEN(x, Antony, Caesar)

Brutus sold Caesar a car

Brutus gave Caesar a car	*Brutus donated a charity a car
Brutus gave a car away	Brutus donated a car
Brutus gave at the office	Brutus donated anonymously

Absent Word Meanings

Striking *absence* of certain (open-class) lexical meanings
that *would* be permitted
if I-Languages permit nonmonadic semantic types

<e,<e,<e,<e, **t**>>>> (instructions to fetch) tetradic concepts

<e,<e,<e, **t**>>> (instructions to fetch) triadic concepts

<e,<e, **t**>> (instructions to fetch) dyadic concepts

<e> (instructions to fetch) singular concepts

Absent Word Meanings

Alexander jimmed the lock a knife

jimmed	➔ JIMMIED(x, z, y)
[jimmed [the lock]	➔ JIMMIED(x, z, the lock)
[[jimmed [the lock] [a knife]]	➔ JIMMIED(x, a knife, the lock)

Brutus froms Rome

froms	➔ COMES-FROM(x, y)
[froms Rome]	➔ COMES-FROM(x, Rome)

Absent Word Meanings

Alexander jimmed the lock a knife

jimmed ➔ JIMMIED(x, z, y)

[jimmed [the lock] ➔ JIMMIED(x, z, the lock)

[[jimmed [the lock] [a knife]] ➔ JIMMIED(x, a knife, the lock)

Brutus talls Caesar

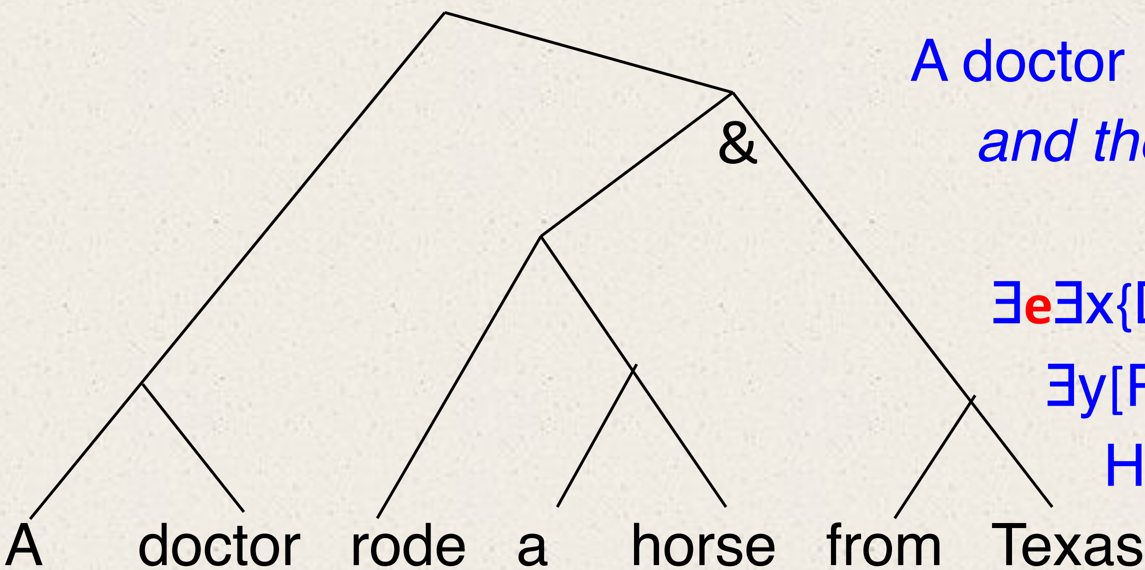
talls ➔ IS-TALLER-THAN(x, y)

[talls Caesar] ➔ IS-TALLER-THAN(x, Caesar)

*Why **doesn't** the structure below support the following meaning:*

A doctor both rode a horse and was from Texas

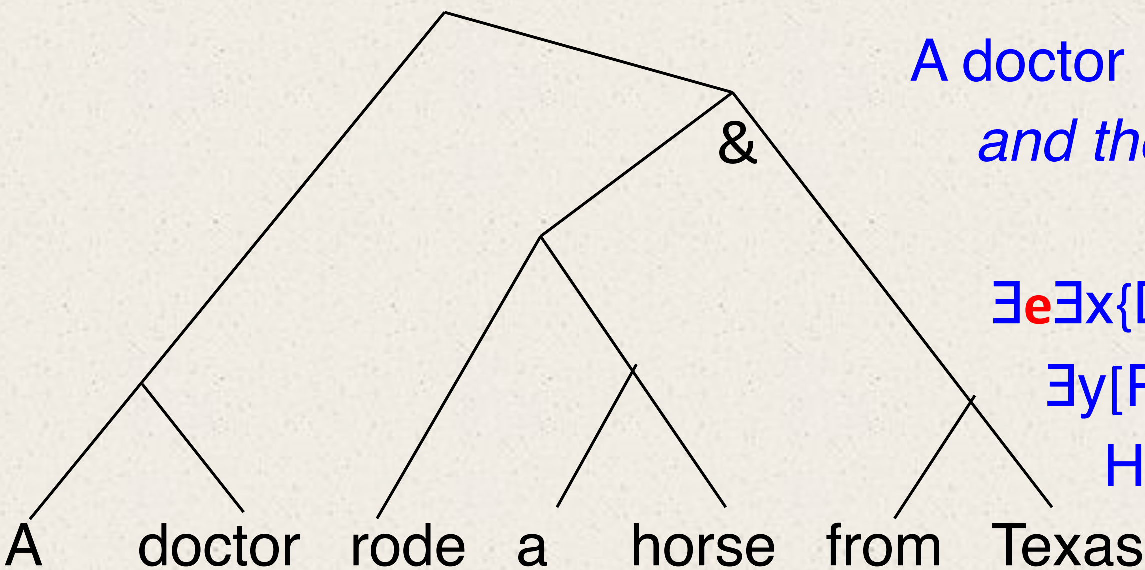
$\exists e \exists \underline{\mathbf{x}}\{\mathrm{Doctor}(\underline{x}) \ \& \ \exists y[\mathrm{Rode}(e, \underline{x}, y) \ \& \ \mathrm{Horse}(y) \ \& \ \mathrm{From}(\underline{x}, \mathrm{Texas})]\}$



A doctor rode a horse *and the ride was* from Texas

$\exists \mathbf{e} \exists x\{\mathrm{Doctor}(x) \ \& \ \exists y[\mathrm{Rode}(e, x, y) \ \& \ \mathrm{Horse}(y) \ \& \ \mathrm{From}(e, \mathrm{Texas})]\}$

Even on Kratzer's view,
the verb 'rode' does not have a
"robustly relational" meaning



A doctor rode a horse
and the ride was from Texas

∃e∃x{Doctor(x) & Agent(e, x)
∃y[Rode(e, y) &
Horse(y) & From(e, Texas)]}

Absent Word Meanings

Striking *absence* of certain (open-class) lexical meanings

that *would* be permitted

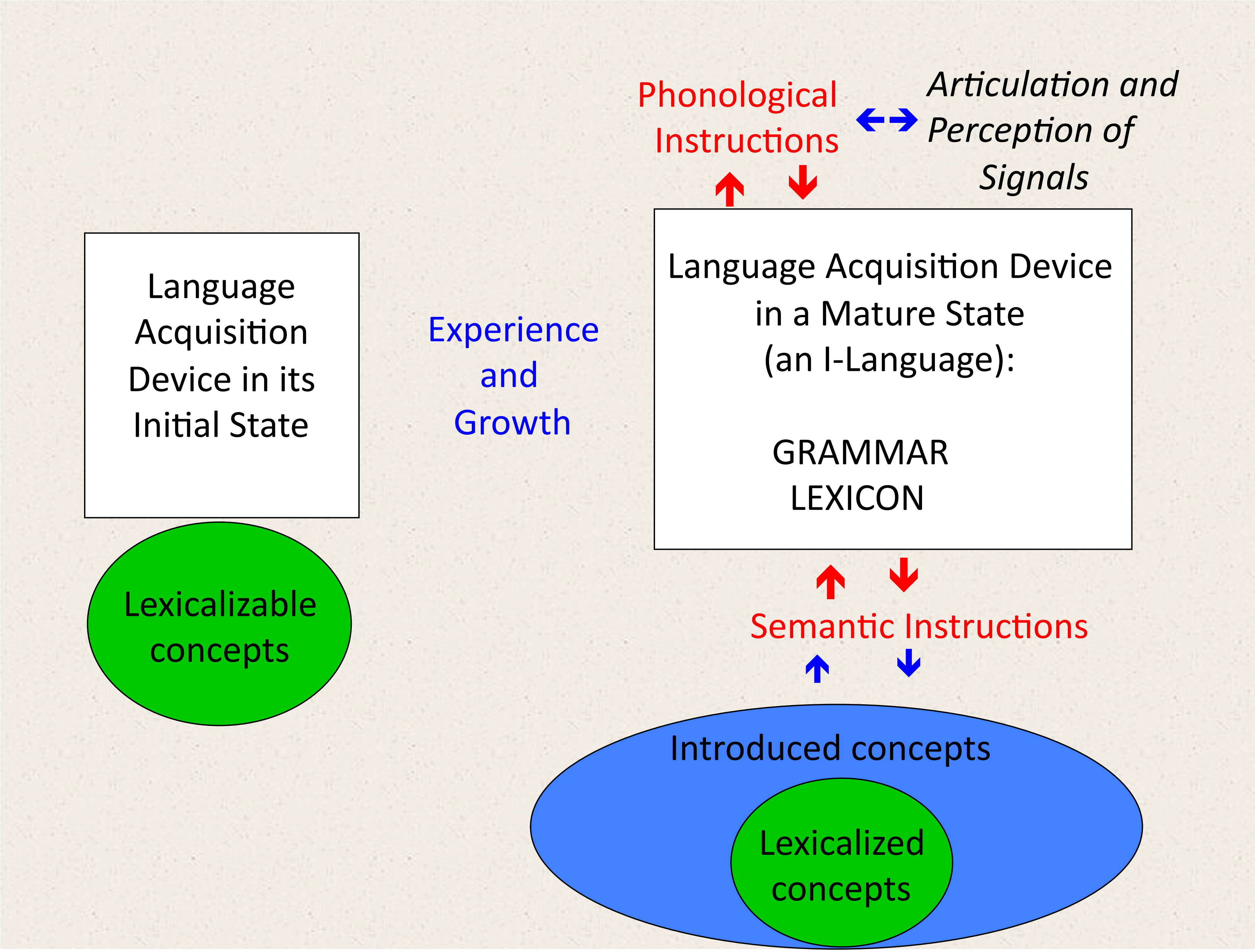
if I-Languages permit nonmonadic semantic types

<e,<e,<e,<e, **t**>>>> (instructions to fetch) tetradic concepts

<e,<e,<e, **t**>>> (instructions to fetch) triadic concepts

<e,<e, **t**>> (instructions to fetch) dyadic concepts

<e> (instructions to fetch) singular concepts


Phonological Instructions
Articulation and Perception of Signals
Language Acquisition Device in its Initial State
Experience and Growth
Language Acquisition Device in a Mature State (an I-Language):
GRAMMAR
LEXICON
Lexicalizable concepts
Semantic Instructions
Introduced concepts
Lexicalized concepts

Back to the Main Idea

- In acquiring words, kids use available concepts to *introduce* new ones.

  Sound('ride') + RIDE(_, _) ==> RIDE(_) + RIDE(_, _) + 'ride'

- Meanings are *instructions* for how to access and combine *i-concepts*

  --lexicalizing RIDE(_, _) puts RIDE(_) at an accessible address

  --introduced concepts can be constituents of (variable-free) *conjunctions* that are formed without a Tarskian ampersand

'fast horse'	FAST( )^HORSE( )
'ride a horse'	RIDE( )^∃[Θ(, _)^HORSE(_)]

Meaning('fast horse') = JOIN{Meaning('fast'), Meaning('horse')}

= JOIN{fetch@'fast'), fetch@'horse')}

Back to the Main Idea

- In acquiring words, kids use available concepts to *introduce* new ones.

Sound('ride') + RIDE(_, _) ==> RIDE(_) + RIDE(_, _) + 'ride'

- Meanings are *instructions* for how to access and combine *i-concepts*

--lexicalizing RIDE(_, _) puts RIDE(_) at an accessible address

--introduced concepts can be constituents of (variable-free) *conjunctions* that are formed without a Tarskian ampersand

'fast horse' FAST( )^HORSE( )

'ride a horse' RIDE( )^∃[Θ(, _)^HORSE(_)]

Meaning('ride a horse') = JOIN{Meaning('ride'), Θ[Meaning('horse')]}

= JOIN{fetch@'ride'), Θ[Meaning('horse')]}

= JOIN{fetch@'ride'), Θ[fetch@'horse']}

Comparison with a More Familiar View

Sound('ride') + RIDE(_, _) ==> λy.λx.T ≡ RIDE(x, y)

Sound('Sadie') + SADIE ==> SADIE

Den:'ride Sadie' = *Den*:'ride'(*Den*:'Sadie') = λx.T ≡ RIDE(x, SADIE)

Den:'from Texas' = λx.T ≡ FROM(x, TEXAS)

Den:'horse' = λx.T ≡ HORSE(x)

Den:'horse from Texas' = ???

Comparison with a More Familiar View

'fast horse' FAST( )^HORSE( )

'ride a horse' RIDE( )^adjust[HORSE(_)]

RIDE( )^∃[Θ(, _)^HORSE(_)]

Sound('ride') + RIDE(_, _) ==> λy.λx.T ≡ RIDE(x, y)

Sound('Sadie') + SADIE ==> SADIE

Den:'ride Sadie' = *Den*:'ride'(*Den*:'Sadie') = λx.T ≡ RIDE(x, SADIE)

Den:'from Texas' = λx.T ≡ FROM(x, TEXAS)

Den:'horse' = λx.T ≡ HORSE(x)

adjust[*Den*:'from Texas'] = λX.T ≡ X(x) = T & FROM(x, TEXAS)

Den:'horse from Texas' = adjust[Den:'from Texas'](Den:'horse')

= λx.T ≡ HORSE(x) & FROM(x, TEXAS)

On my view, *meanings* are neither extensions nor concepts.

Meanings are *composable instructions* for how to build concepts.

So the meaning of 'horse' is a *part* of the meaning of 'fast horse'.

Meaning('fast') = fetch@'fast')

Meaning('horse') = fetch@'horse')

Meaning('fast horse') = JOIN{Meaning('fast'), Meaning('horse')}

= JOIN{fetch@'fast'), fetch@'horse')}

But "instructionism" and "conjunctvism" are distinct theses

Meaning('ride Sadie') = APPLY{Meaning('ride'), Meaning('Sadie')}

= APPLY{fetch@'ride'), fetch@'Sadie')}

L is "Semantically Compositional" if...

(A) at least some expressions of L have "semantic values"
that can be specified in terms of finitely many

-- *lexical axioms* that specify the semantic values of *atomic* L-expressions, and

-- *phrasal axioms* that specify the semantic values of *complex* L-expressions
in terms of the semantic values of their (immediate) constituents

(B) each expression of L *has a meaning*
that is *constituted* by the meanings of its (immediate) constituents

lexical axioms describe the meanings of *atomic* L-expressions
in a way that encodes the *typology* required by the *phrasal axioms*,
which describe how the meanings of *atomic* L-expressions are *built*

The Meaning of Merging: Restricted Conjunction

if M is a monadic concept with which we can *think about* Ms

and C is a monadic concept with which we can *think about* Cs,

then C^M is a conjunctive monadic concept with which

we can think about Ms that are also Cs

RED^BARN( ) applies to e iff both BARN( ) and RED( ) apply to e

The Meaning of Merging: Restricted Conjunction/Closure
(*allowing for a smidgeon of dyadicity*)

if M is a monadic concept with which we can *think about* Ms

and D is a dyadic concept with which we can

think about things that are D-related to other things,

then D^M is a conjunctive monadic concept with which we can

think about things that are D-related to an M

INTO^BARN( ) applies to e iff for some e',

BARN( ) applies to e', and

INTO(,) applies to <e, e'>

- Predicate-Argument:
  Francois saw Pierre — Francois saw Pierre ride horses
- Predicate-Adjunct:
  ride fast — fast horse
- Relative-Clauses:
  what Francois saw — who saw Pierre
- Quantifier+Restrictor
  every horse — most horses
- RestrictedQuantifier+Scope
  every horse saw Pierre — Pierre saw every horse

Predicate-Argument:

Francois saw Pierre

Francois saw Pierre ride horses

Higginbotham: ***Θ-linking***

Θ2(e, Francois) & Saw(e, 2, 1) & Θ(e, Pierre)

Θ2(e’, Pierre) & Ride(e’, 2, 1) & Θ(e’, sm horses)

Θ2(e, Francois) & Saw(e, 2, 1) & Θ(e, sm[Pierre ride sm horses])

Heim/Kratzer: ***function-application*** (with ‘e’-variables)

[[λy.λx.λe.T iff Saw<e, x, y>(Pierre)](Francois)]

Saw<e, F, P> ➔ Θ2<e, F> & Saw<e, P>

[[λy.λx.λe.T iff Saw<e, x, y>(sm[Pierre ride sm horses])](Francois)]

Predicate-Argument:

Francois saw Pierre

Francois saw Pierre ride horses

Higginbotham: *Θ-linking*

Θ2(e, Francois) & Saw(e, 2, 1) & Θ(e, Pierre)

Θ2(e’, Pierre) & Ride(e’, 2, 1) & Θ(e’, sm horses)

Θ2(e, Francois) & Saw(e, 2, 1) & Θ(e, sm[Pierre ride sm horses])

Proposed Variant

∃[Θ2(, _)^THAT-FRANCOIS(_)]^SAW( )^∃[Θ(, _)^THAT-PIERRE(_)]

∃[Θ2(, _)^THAT-PIERRE(_)]^RIDE( )^∃[Θ(, _)^HORSES(_)]

∃[Θ2(, _)^THAT-FRANCOIS(_)]^SAW( )^∃[Θ(, _)^...(_)]

Human Language: a language that human children can naturally acquire

(D) for each *human* language, there is a theory of *truth* that is also
the core of an adequate theory of *meaning* for that language

(C) each human language is an i-language:
a biologically implementable *procedure that generates*
expressions that connect meanings with articulations

(B) each human language is an i-language for which
there is a theory of truth that is also
the core of an adequate theory of meaning for that i-language

(D) for each *human* language, there is a theory of *truth* that is also the core of an adequate theory of *meaning* for that language

Good Ideas	Bad Companion Ideas
"e-positions" allow for conjunction reductions	"e-positions" are Tarskian variables that have mind-independent values
as Foster's Problem reveals, humans *compute* meanings via specific operations	the meanings computed are truth-theoretic properties of *human i-language* expressions
Liar Sentences don't preclude meaning theories for human i-languages	Liar T-sentences are true ('The first sentence is true.' iff the first sentence is true.)

(D) for each *human* language, there is a theory of *truth* that is also the core of an adequate theory of *meaning* for that language

Good Ideas

"e-positions" allow for conjunction reductions

as Foster's Problem reveals, humans *compute* meanings via specific operations

Liar Sentences don't preclude meaning theories for human i-languages

Bad Companion Ideas

characterizing meaning in truth-theoretic terms yields good analyses of specific constructions

such characterization also helps address foundational issues concerning how human linguistic expressions could exhibit meanings at all

Main Idea: Short Form

- In acquiring words, kids use available concepts to *introduce* new ones.

  Sound('ride') + RIDE(_, _) ==> RIDE(_) + RIDE(_, _) + 'ride'

- Meanings are *instructions* for how to access and combine *i-concepts*

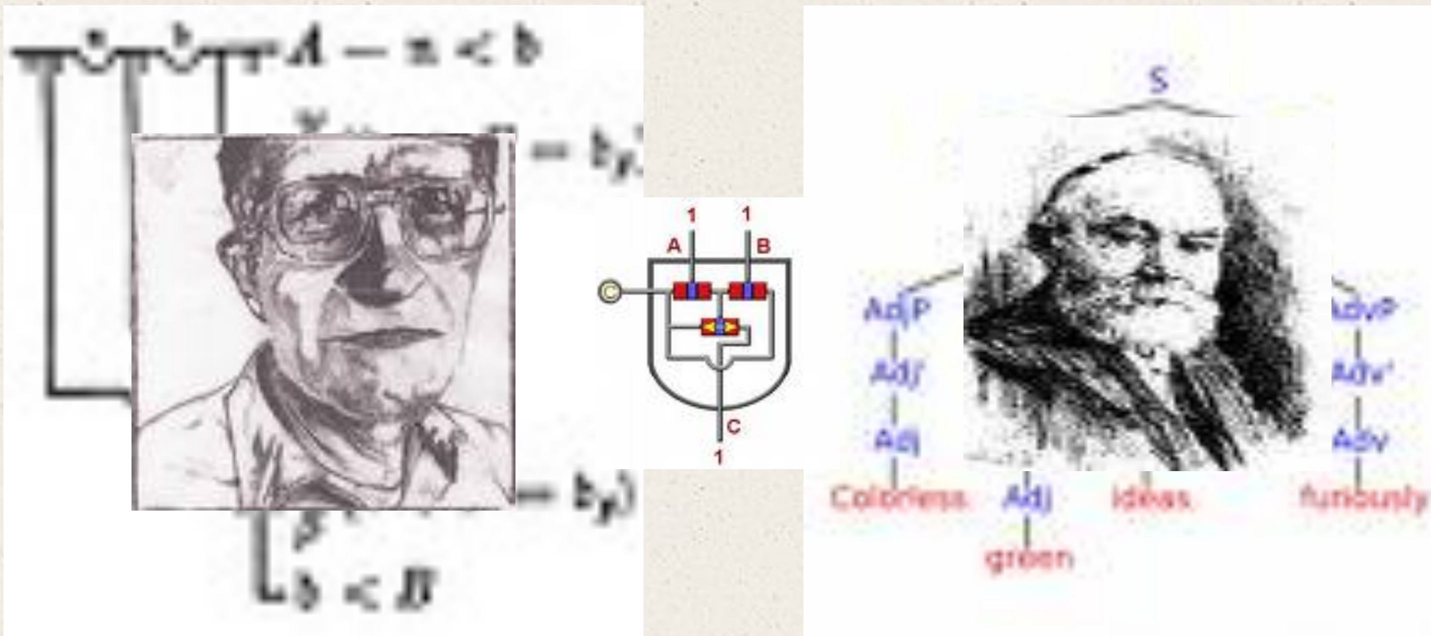
  -- lexicalizing RIDE(_, _) puts RIDE(_) at an accessible address

  -- introduced concepts can be constituents of (variable-free) *conjunctions* that are formed without a Tarskian ampersand

'fast horses' FAST( )^HORSES( )

'ride horses' RIDE( )^∃[Θ(, _)^HORSES(_)]

Meanings First



MANY THANKS

- Predicate-Argument:

  Francois saw (a/the/every) Pierre   Francois saw Pierre ride horses

  does *saturation/function-application/Θ-linking* do any work not done by thematic concepts and simple forms of conjunction/∃-closure?

- Predicate-Adjunct: ride fast   fast horse

  here, everybody appeals to a simple form of conjunction

  Higginbotham: *Θ-binding*

  Heim & Kratzer: *Predicate Modification*

- Relative-Clauses: what Francois saw

  here, everybody appeals to a syncategorematic abstraction principle

  one way or another: Francois saw A1 ➔

  for some A' such that A' $\approx_1$ A, Francois saw A'1

Quantifier+Restrictor	RestrictedQuantifier+Scope
every horse	every horse saw Pierre
	Pierre saw every horse

(1) Saturation + RestrictedAbstraction

every horse	[λY.λX.T iff EVERY<X, Y>(λx.T iff Horse(x)]
Pierre saw Sadie	T iff ∃e[Saw(e, Pierre, Sadie)]
Pierre saw _	λx.T iff ∃e[Saw(e, Pierre, x)]
every horse [Pierre saw _]	EVERY< , λx.T iff Horse(x)>
every horse [who Pierre saw _]	

So why *doesn't* Every horse who Pierre saw have a sentential reading?

And if determiners express relations, why are they conservative?

Quantifier+Restrictor	RestrictedQuantifier+Scope
every horse	every horse saw Pierre
	Pierre saw every horse

(1) Saturation + RestrictedAbstraction

(2) Conjunction/∃-closure/ThematicConcepts + RestrictedAbstraction

Francois saw Pierre

∃[External(, _)^That-F(_)]^Saw( )^∃[Internal(, _)^That-P(_)]

That$_2$GuySawThat$_1$Guy( )

∃e[That$_2$GuySawThat$_1$Guy(e)]

↑-That$_2$GuySawThat$_1$Guy( )

1[↑-That$_2$GuySaw~~Which$_1$ Person~~( )

for some A’ such that A’ $\approx_1$ A, A’2 saw A’1

Lots of Conjoiners, Semantics

- If π and π* are propositions, then

  TRUE(π & π*) iff TRUE(π) *and* TRUE(π*)

- If π and π* are monadic predicates, then for each entity x:

  APPLIES[(π $\&^{M}$ π*), x] iff APPLIES[π, x] *and* APPLIES[π*, x]

- If π and π* are dyadic predicates, then for each ordered pair o:

  APPLIES[(π $\&^{DA}$ π*), o] iff APPLIES[π, o] *and* APPLIES[π*, o]

- If π and π* are predicates, then for each sequence σ:

  SATISFIES[σ, (π $\&^{PA}$ π*)] iff SATISFIES[σ, π] *and* SATISFIES[σ, π*]

  APPLIES[σ, (π $\&^{PA}$ π*)] iff APPLIES[π, σ] *and* APPLIES[π*, σ]