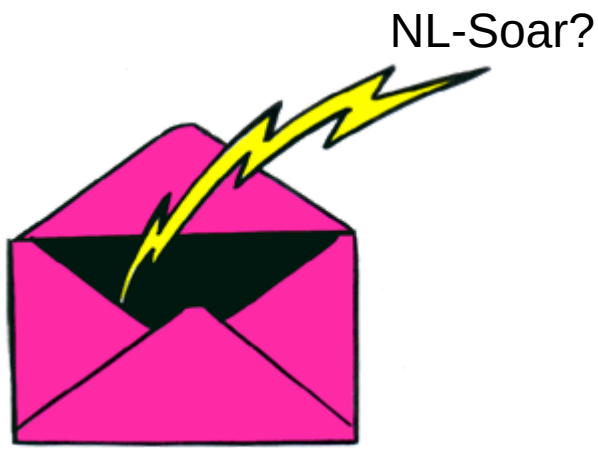
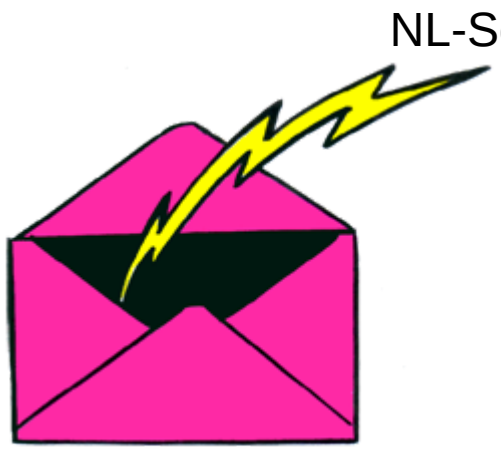
Progress on NL-Soar, and Introducing XNL-Soar

Deryle Lonsdale, Jamison Cooper-Leavitt,
and Warren Casbeer
(and the rest of the BYU NL-Soar Research Group)
BYU Linguistics
lonz@byu.edu

What appears to be happening with NL-Soar support

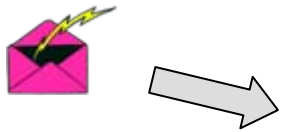


What appears to be happening with NL-Soar support



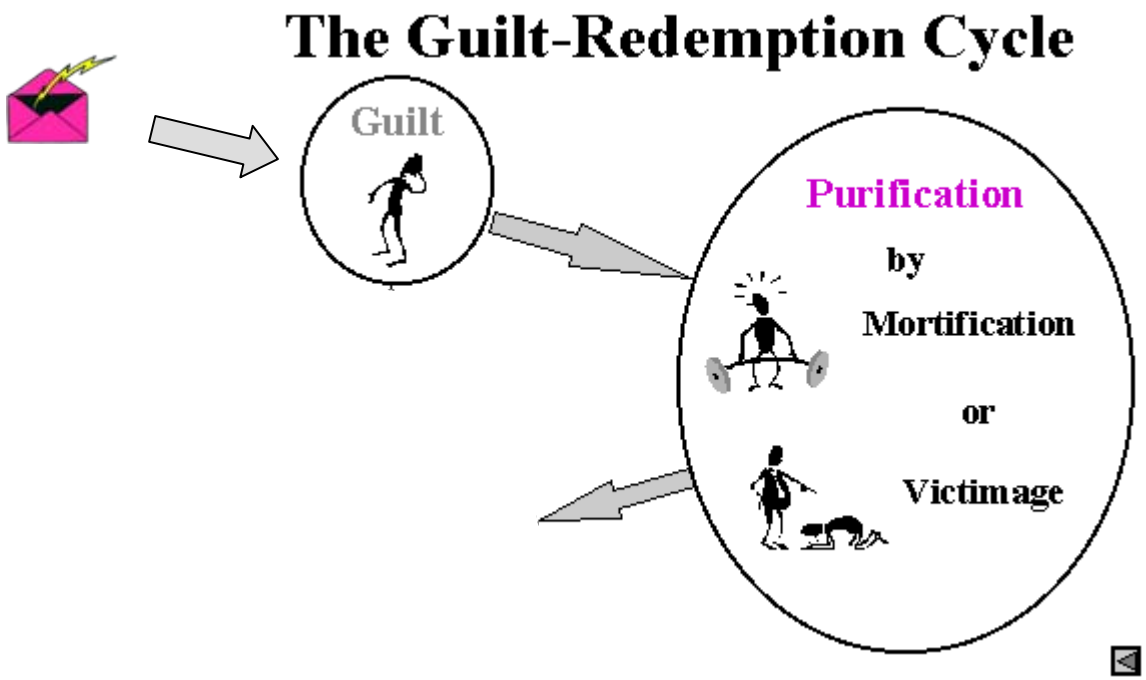
> /dev/null

What's actually happening with NL-Soar support

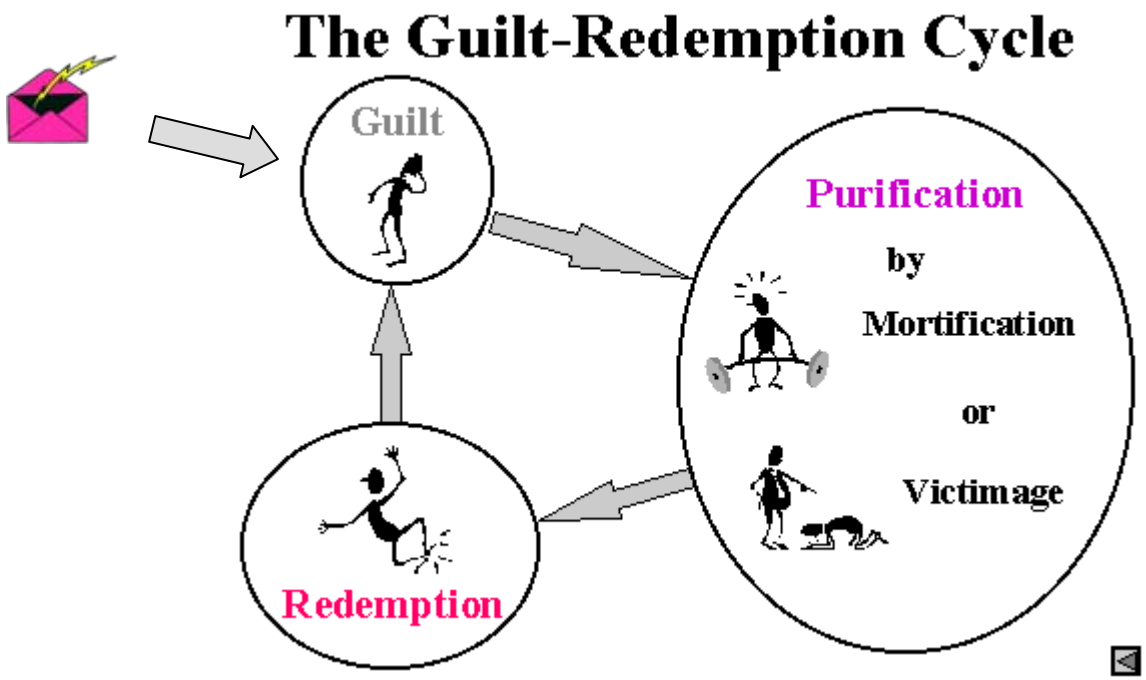


What's actually happening with NL-Soar support

What's actually happening with NL-Soar support



What's actually happening with NL-Soar support



NL-Soar developments (1)

- Discourse/robotic dialogue
  - Sphinx-4 speech input (working on lattice-based interface)
  - Festival text-to-speech output
  - Two agents holding a (short) conversation
  - Video produced showing round-trip speech-based human/robot interaction
  - NSF proposal submitted

NL-Soar developments (2)

- NL generation
  - Decoupled from comprehension
  - Can be driven from arbitrary LCS
    - Front-end GUI for creating LCS's
- Port to Soar 8.5.2
  - Some NLG chunking issues remain
- Modeling of cognition in simultaneous interpretation (English-French)

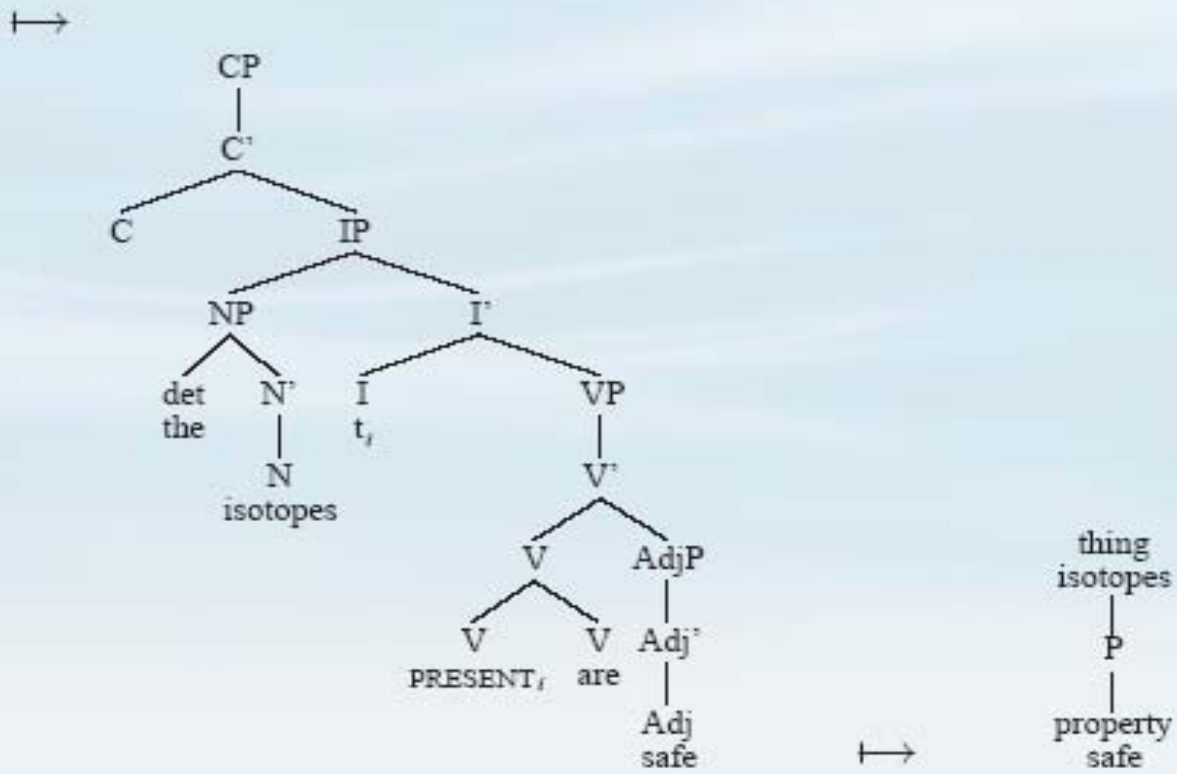
SI from a cognitive modeling perspective

SI shares some interesting characteristics with agent modelling and simulation systems.

- Interpreter is an agent in a highly volatile environment (linguistically speaking)
- Split-second control decisions must be taken
- Several subtasks must be managed/interleaved (listening, translating, speaking)
- Finite cognitive resources must be managed
- Some mixture of deliberate/learned behavior

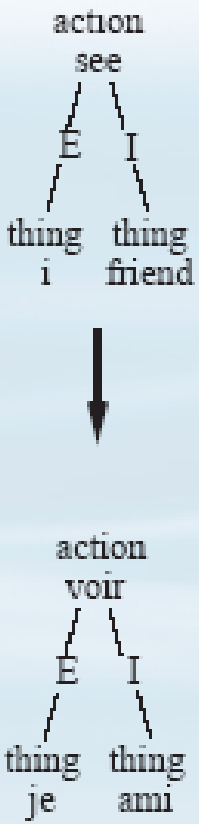
Parsing and the models

The ... isotopes ...
are ... safe...



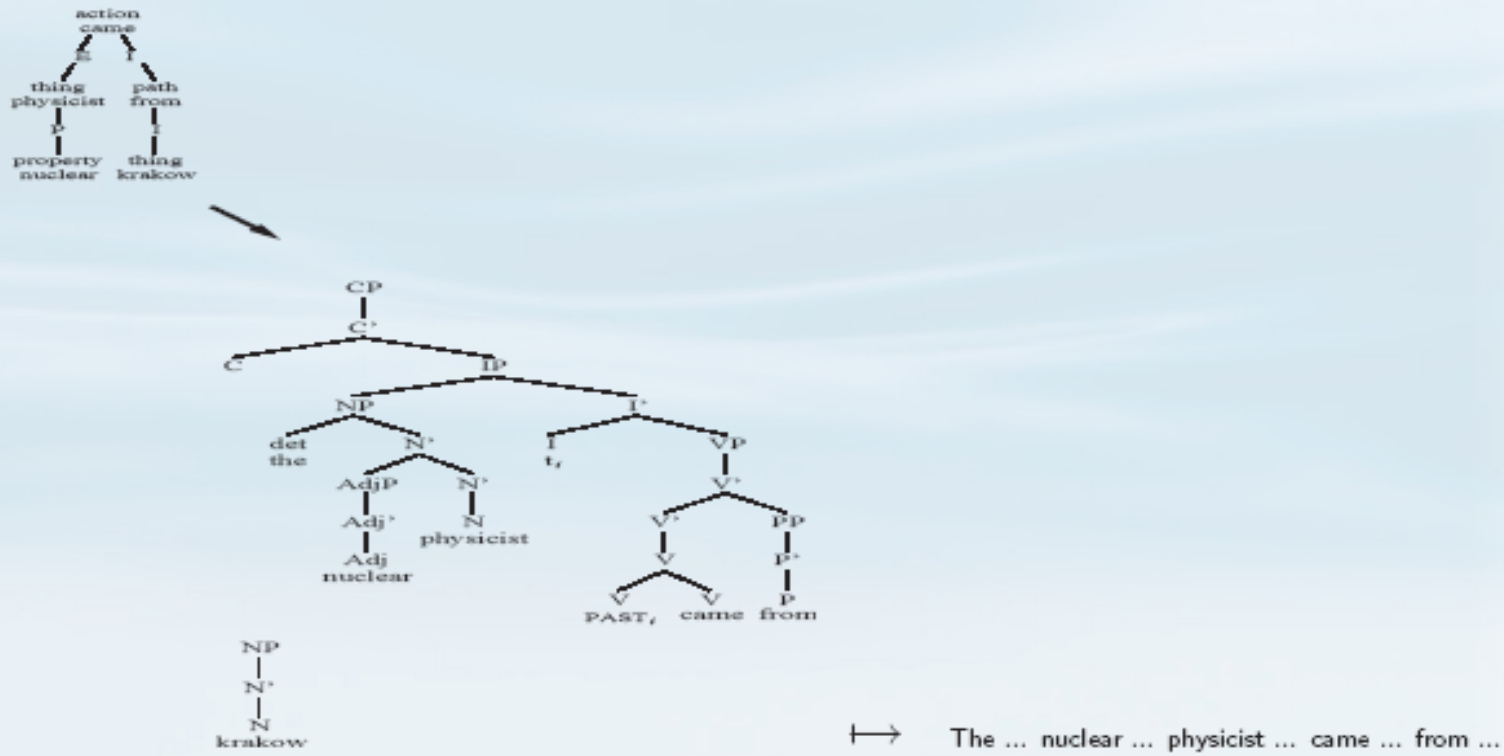
Mapping operators

Incrementally map SL semantic model to TL semantic model.

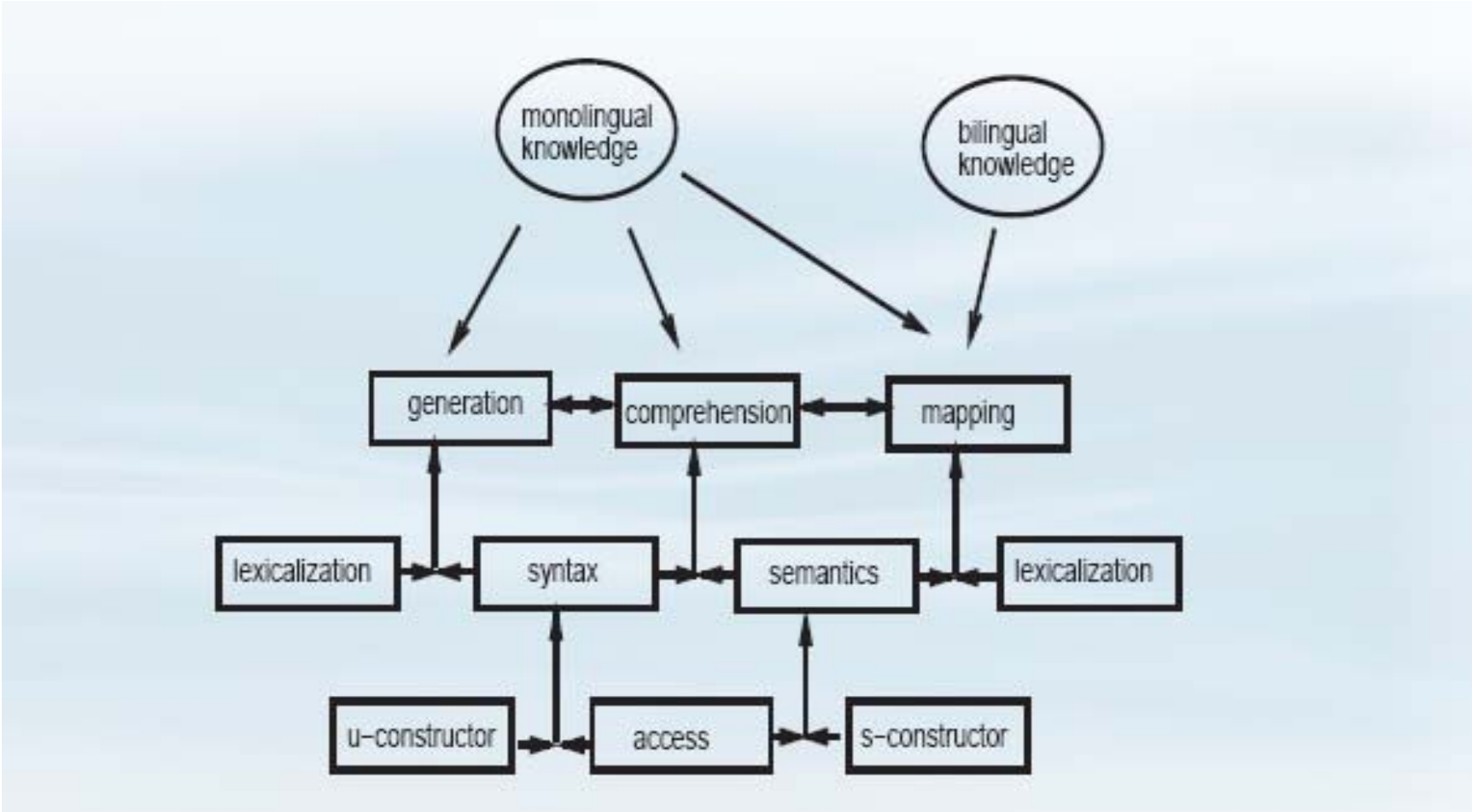


NL-Soar generation operators

Incrementally convert semantic structure to syntactic trees and then output sentences via s-realize operators.



Combining the capabilities

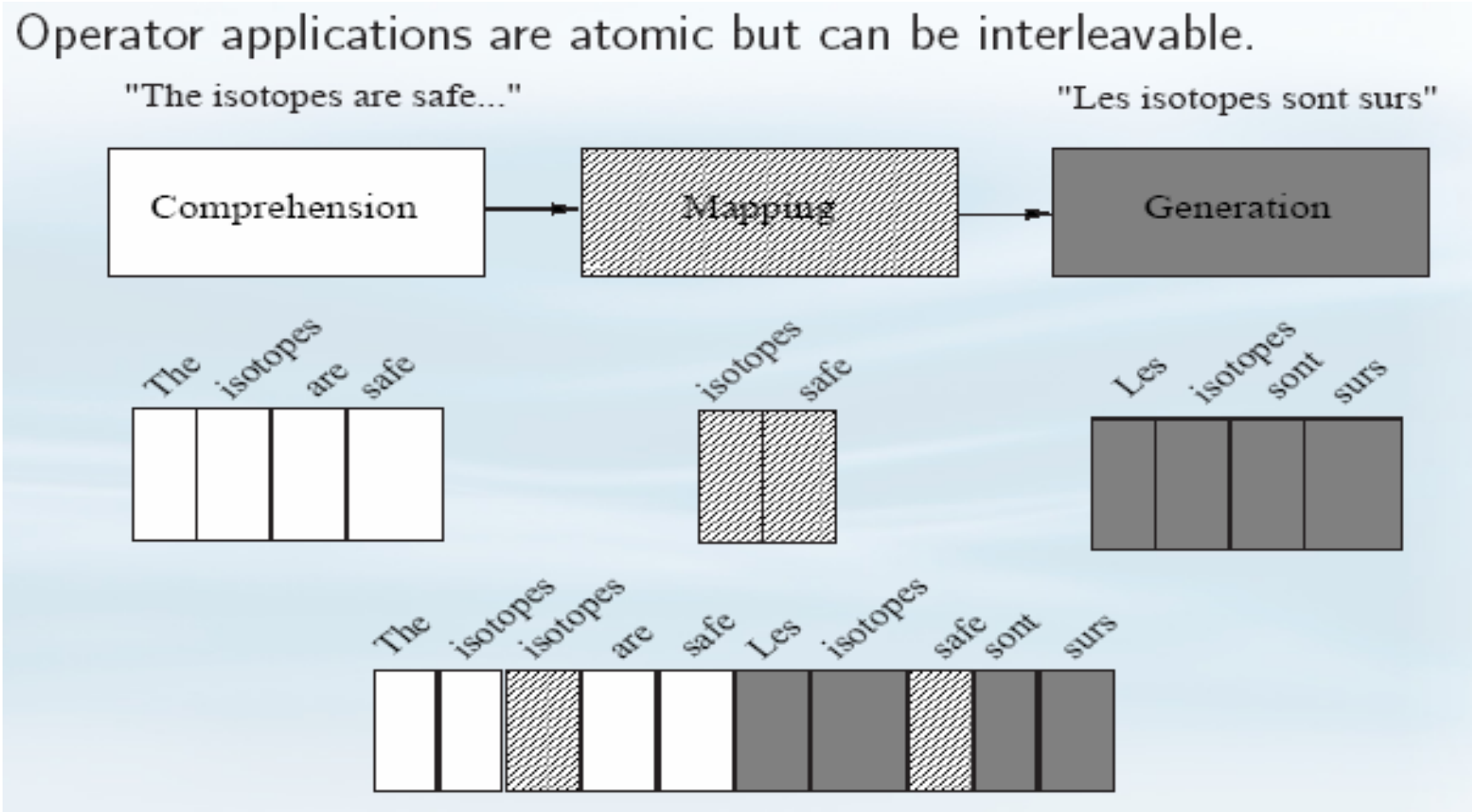


Pipelining the processes

Linear (unlikely) scenario for interpreting "The isotopes are safe."

Operation	Operator
attend to "the"	access
build syntax for "the"	u-cstr
attend to "isotopes"	access
build syntax for "isotopes"	u-cstr
build semantics for "isotopes"	s-cstr
attend to "are"	access
build syntax for "are"	u-cstr
attend to "safe"	access
build syntax for "safe"	u-cstr
build semantics for "safe"	s-cstr
lexicalize, access, and build Frn concept for "isotopes"	m-cstr
lexicalize, access, and build Frn concept for "safe"	m-cstr
lexicalize, access, and build Frn definite article	s-realize
lexicalize, access, and build Frn noun "isotopes"	s-realize
lexicalize, access, and build Frn copula	s-realize
lexicalize, access, and build Frn adjective "sûrs"	s-realize

Interleaving operator implementations



Interleaving the processes

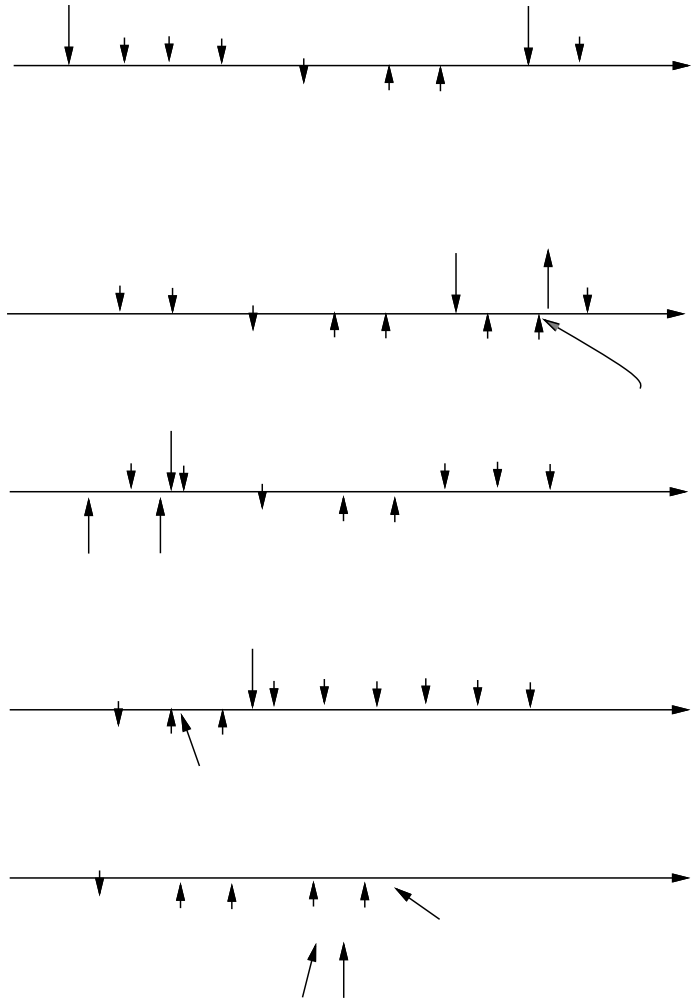
Interleaved scenario for interpreting "The isotopes are safe."

Operation	Operator
attend to "the"	access
build syntax for "the"	u-cstr
attend to "isotopes"	access
build syntax for "isotopes"	u-cstr
build semantics for "isotopes"	s-cstr
map English concept "isotopes" into French	m-cstr
generate definite article for concept "isotopes"	s-realize
attend to "are"	access
build syntax for "are"	u-cstr
generate French noun "isotopes"	s-realize
generate copula	s-realize
attend to "safe"	access
build syntax for "safe"	u-cstr
build semantics for "safe"	s-cstr
map English concept "safe" into French	m-cstr
generate French adjective "sûrs"	s-realize

Predicted times by operator type

	Number	Time (msec)
lexical access	6	300
u-constructor	6	300
s-constructor	6	300
select (mapping)	5	250
m-constructor	5	250
select(generation)	7	350
s-realize	7	350
say	7	350
Total	49	2450

Event timeline (one possibility)

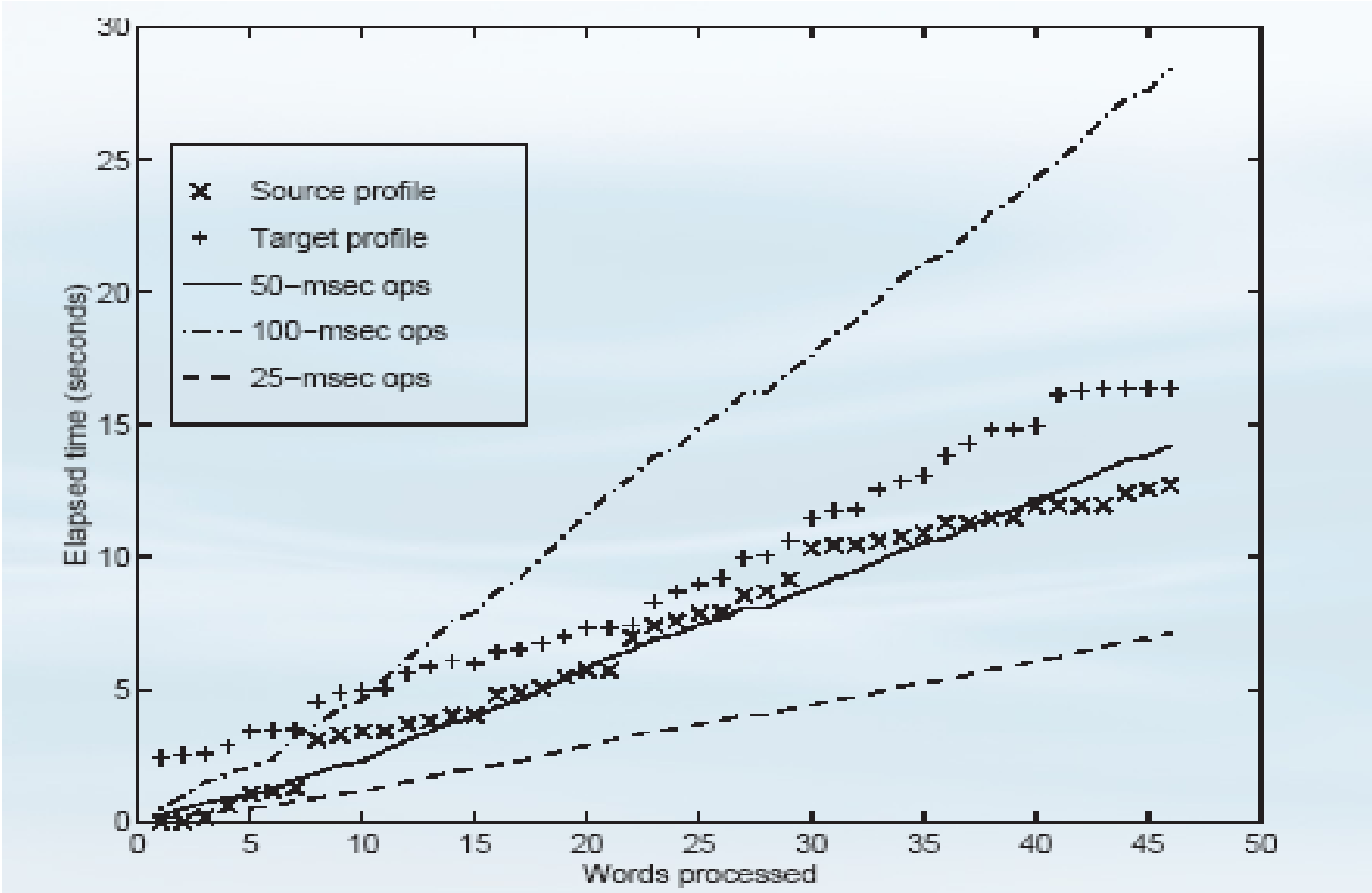


Sample alignment analysis

Word	Onset	Coda	Total	Word	Onset	Coda	Total
I	2884	2939	55	J+	3035	3040	5
VISITED	2940	2984	44	AI	3041	3047	6
ALEX	2985	3028	43	VISITE1	3048	3098	50
AT	3029	3037	8	ALEX	3099	3162	63
THE	3038	3049	11	A2	3163	3165	2
HOSPITAL	3050	3107	57	L+	3166	3170	4
+PAUSE+	3108	3230	122	HO3PITAL	3171	3218	47

- Divide time duration (2.45 seconds) by posited operators
- There should be 49 operators
- Hence 49! possible unconstrained operator sequences
- Constraints: time per operator, time course, precedence of operators

Observed profile and timing assumptions

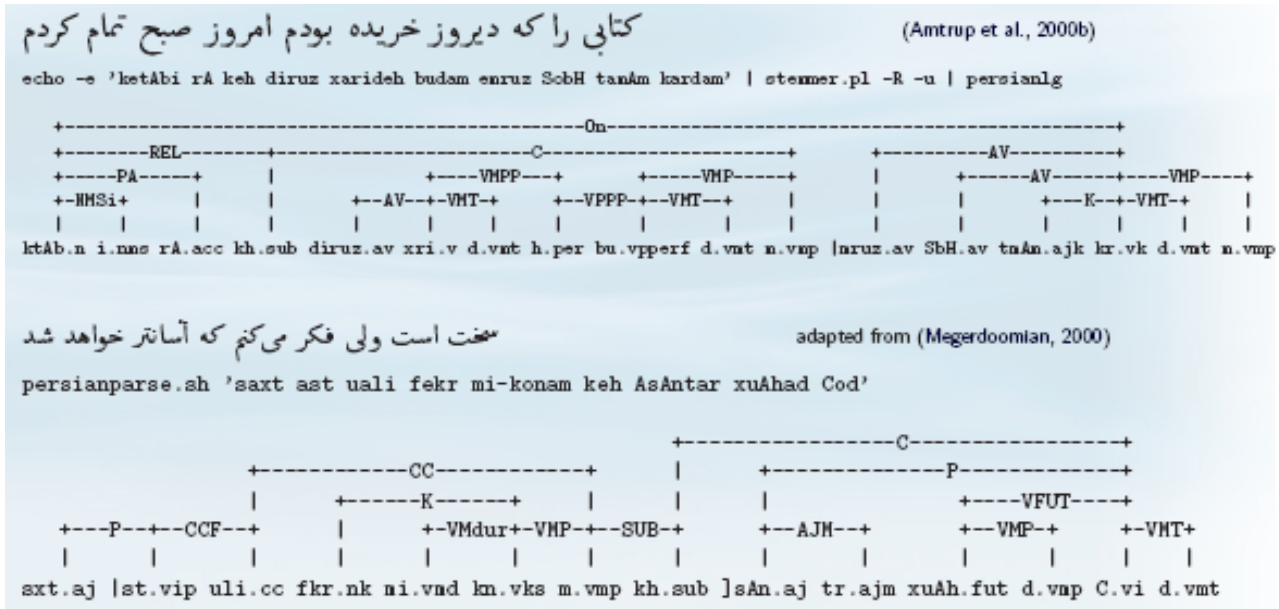


First 1/3 of an interleaved scenario timeline

Acrobat Document

LG-Soar developments

- Predicate extraction in biomedical texts domain (www.clinicaltrials.gov)
- Scaling up of Persian syntactic parser



Unveiling XNL-Soar: Minimalism and Incremental Parsing

What are we trying to do?

- As with NL-Soar, study how humans process language
  - Lexical access
  - Syntax/semantics
- Apply the Soar architecture
  - Operator-based cognitive modeling system
  - Symbolic, rule-based, goal-directed agent
  - Learning
- Implement syntax in the Minimalist Program

Why XNLS? (1)

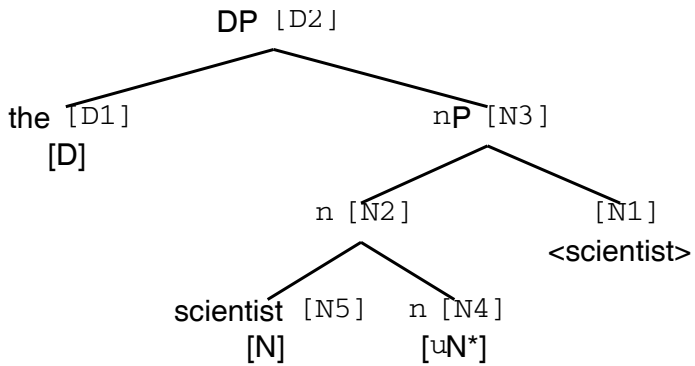
- GB has been (largely) superseded by MP
- It's a debatable development (e.g. recent LinguistList discussion/flamefest)
- No large-scale MP parser implemented yet
- No MP generator implemented yet
- Flavor seems right (even operators!)

I just re-read Rick's thesis, and I wondered if you've thought at all about applying "newer" grammars (e.g., Chomsky's "minimalist programme") in NL-Soar? (Chris Waterson, June 17, 2002)

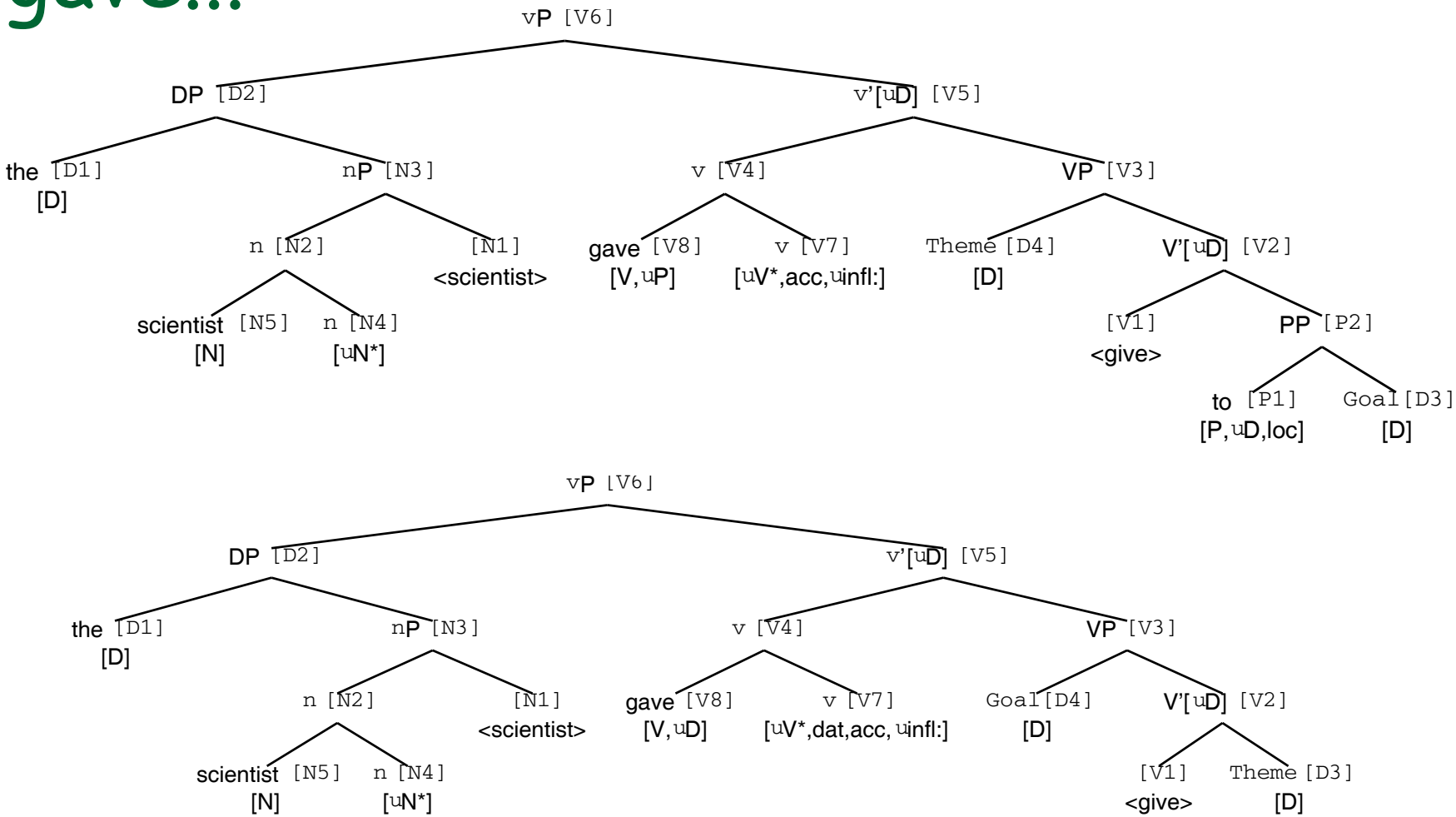
Why XNLS? (2)

- Incrementality of MP not explored
- Unknown whether MP viable for human sentence processing (but claimed to be)
- Experience with another formalism
  - Syntax so far: GB, Link Grammar
  - Semantics so far: Annotated models, LCS, DRT
- Pedagogical aims

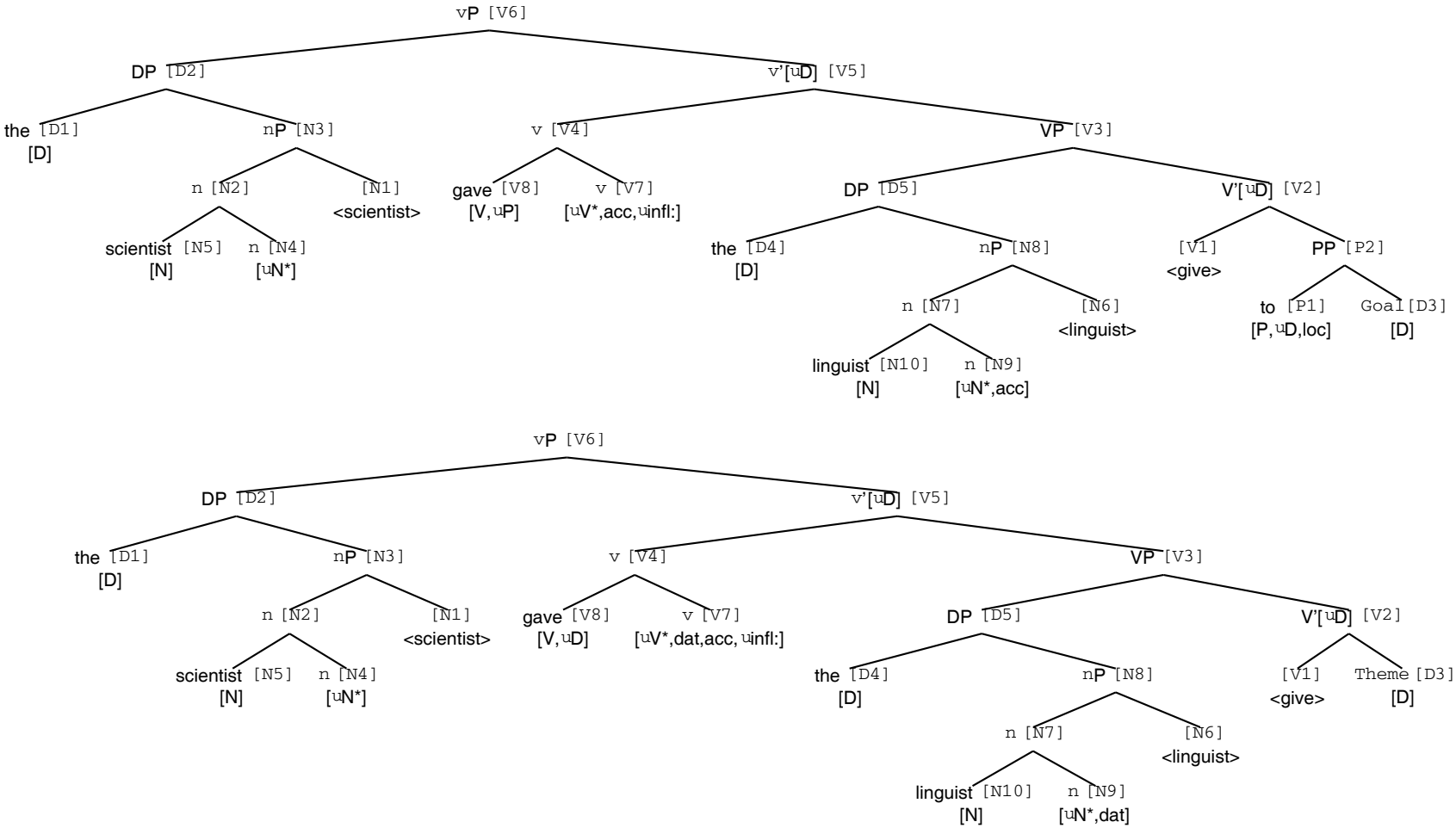
After hearing "The scientist..."



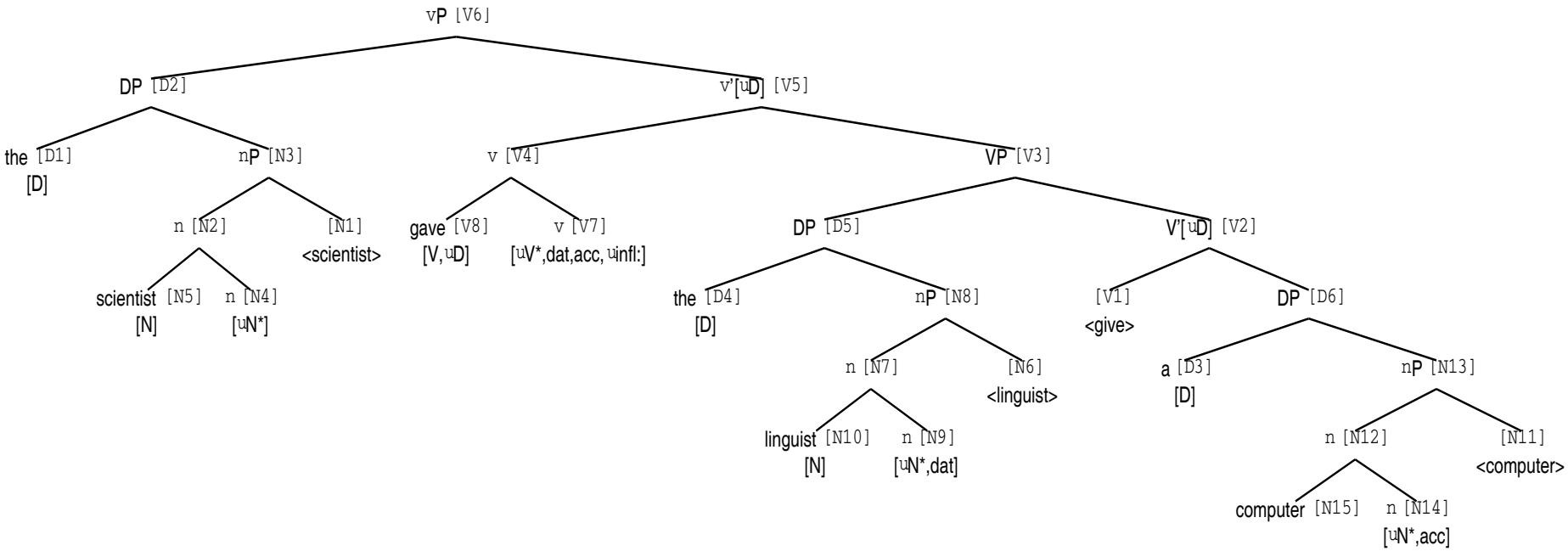
After hearing "The scientist gave..."



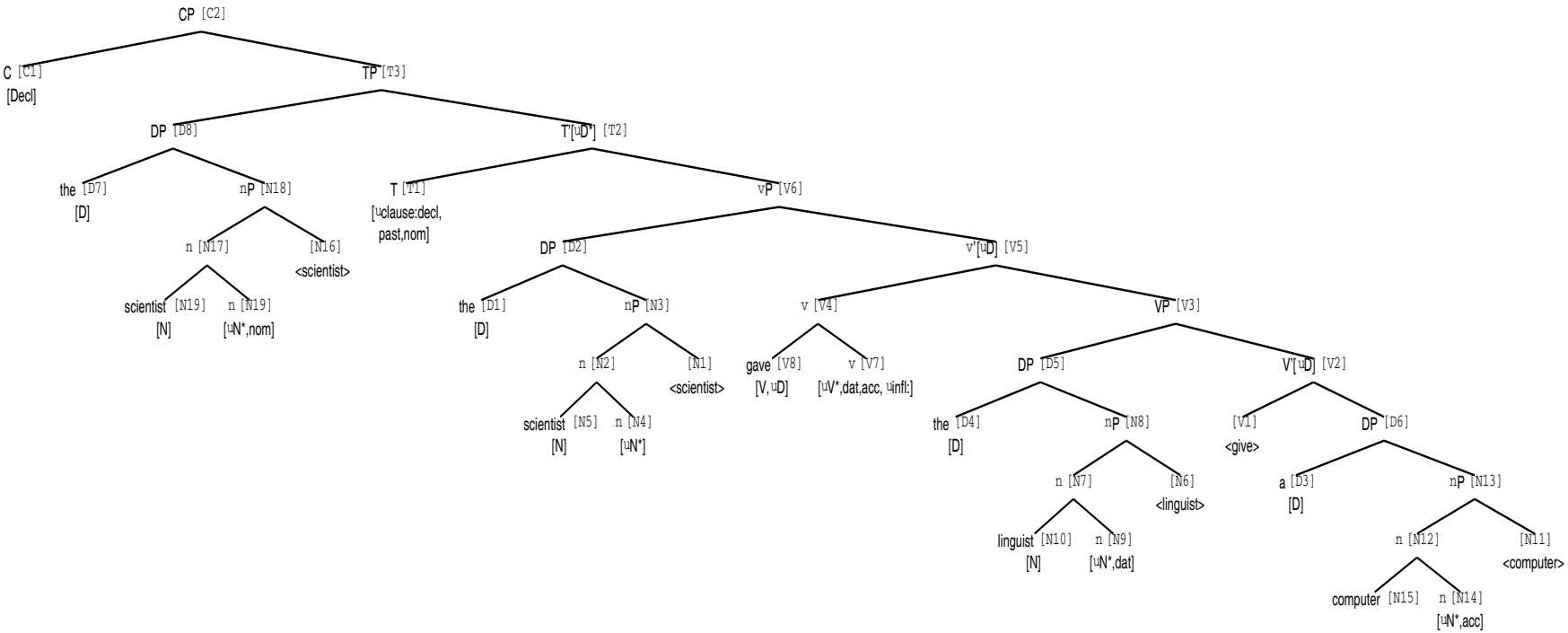
After hearing "The scientist gave the linguist…"



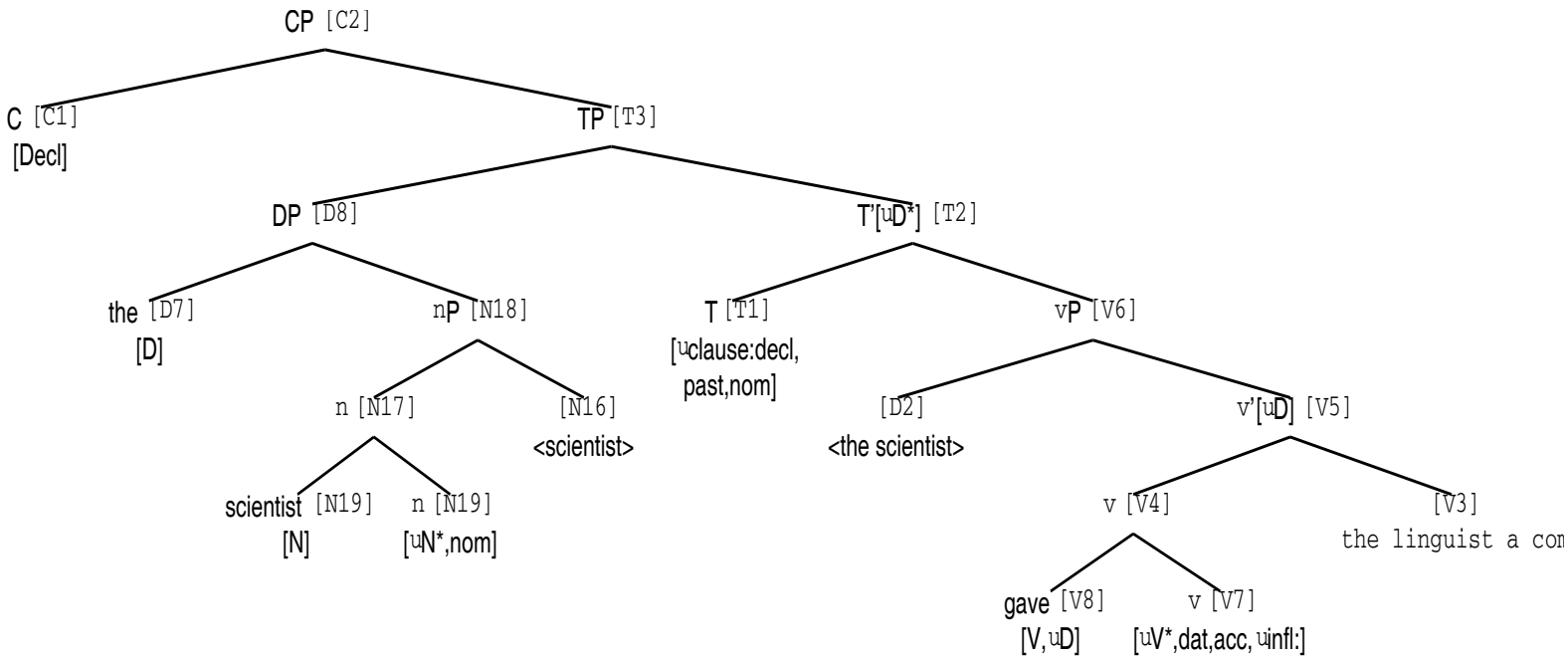
After hearing "The scientist gave the linguist a computer."



Projecting the structure



Completed tree for "The scientist gave the linguist a computer."



Operator types (still to be done)

- (Attention)
- Lexical access (from NL-Soar, including WordNet)
- Merge: link 2 pieces of syntactic structure
  - Constraints: subcategorization, (hierarchy of projections), (theta roles: PropBank?)
- Move: moves a constituent (e.g. questions)
  - Constraints: locality, features
- (Snip)

Operator types

- Inherit from NL-Soar:
  - Semantics: build pieces of conceptual representation
  - Discourse: select and instantiate discourse plans for comprehension and generation
  - Generation: generate text from semantic representation

Other system components

- Assigners/receivers set?
- Parameterized decay-prone I/O buffer
- New grapher for MP parse trees

Current status

- Current XNLS system: about 40 rules (c.f. NL-Soar system: 3500 rules)
- Intransitive sentences
  - Basic sentences work (e.g. 'zebras sneezed')
- WordNet gives us uninflected forms; this is a problem for generation

Expected payoffs

- Crosslinguistic development
  - Easier to parameterize due to features
- Wider coverage of complex constructions
  - Ditransitives, resultatives, causatives, unaccusatives, etc.
- More workable platform for implementing partial analyses from the literature

Conclusion

- Coals
  - Performance?
  - MP not fully explored
  - More highly lexicalized, so more lexical resources required
  - XNLS entails the Guilt-Redemption cycle
- Nuggets
  - Better coverage (Engl. & crosslinguistically)
  - New start in Soar8
  - State-of-the-art syntax
  - Puts us in the thick of the battle
  - Relevance to current linguistic pedagogy