Attempto Controlled English: Language, Tools and Applications

The Language ACE

Norbert E. Fuchs, Kaarel Kaljurand

Department of Informatics & Institute of Computational Linguistics
University of Zurich

{fuchs, kalju}@ifi.unizh.ch
http://attempto.ifi.unizh.ch

December 2006

The Language ACE

- Like every language, ACE has
  - vocabulary/morphology
  - syntax
  - semantics
  - pragmatics
- and one must learn it.
- If you know English then you already know most of it.
- So we focus on how ACE differs from English.

English and ACE

- English is both syntactically and semantically more powerful than ACE.
- English is used in human-human communication while ACE is also meant to be used in human-computer communication…
- which means that ACE is really a formal language (like various logics), it simply resembles English syntactically and semantically.

English and ACE

- Like English, ACE allows syntactically different sentences to express the same meaning (i.e. there is a lot of synonymy)
- Unlike English, each sentence is interpreted in just one way (i.e. there is no ambiguity)
  - anaphoric dependencies are an exception
- I.e. in ACE the smallest unambiguous unit is a sentence while in English the ambiguity is resolved by much more distant dependencies

Learning ACE

- Vocabulary/morphology
  - predefined *function words* (articles, prepositions, …)
  - predefined *fixed phrases* ('there is a ...', 'it is false that ...')
  - user-defined *content words* (nouns, verbs, adjectives, adverbs) and their forms
- Construction rules (i.e. syntax)
  - define admissible sentence structures
  - avoid ambiguous or imprecise constructions
- Interpretation rules (i.e. semantics)
  - control logical analysis of admissible sentences
  - resolve remaining ambiguities
- Style guide, tools (i.e. pragmatics)

Vocabulary

- Content words
  - dog, cat, like, manually, red, …
- Function words
  - if, who, 42, in, not, of, …
- Words can change morphologically
  - 1 **man**, 2 **men**
  - **likes**, does not **like**, is **liked** by
  - **good**, **better** than, **best**

ACE Content Words

- The ACE content words are adverbs, adjectives, nouns and verbs
- The set of ACE content words is infinitely large and dynamic
- All English content words are also ACE content words
  - e.g. podcast, groovy
- Multiword units are always hyphenated
  - e.g. fill-in
  - e.g. persona-non-grata

Semantics of Content Words

- There is no predefined lexical semantics
  - e.g. *bank* is underspecified
  - e.g. *narcissist* is just a noun with no relation to the verb *like*
  - e.g. *unacceptable* is just an adjective with not encoded negation in it
- One can of course add this semantics by using ACE sentences
  - Every narcissist likes himself.
- This information however would not be used during parsing, but only by an eventual reasoning tool.

ACE Adverbs and Adjectives

- Positive, comparative and superlative adverbs
  - quickly, more quickly, most quickly
- Positive, comparative and superlative adjectives
  - tall, taller than somebody, tallest
- Adjectives can have a PP-object
  - fond-of something

ACE Nouns

- Common nouns, propernames, measurement nouns
  - man, John, kg
- Strings and numbers (integers)
  - "Go!", 12, -3
- Singular and plural forms
  - man, men, John, Christmases
- Countable and mass nouns
  - man, money, laundry
- Gender (masculine, feminine, neutrum)
  - John, Mary, table

ACE Verbs

- Phrasal and prepositional verbs
  - drop-out, fill-in
- Subcategorization (intransitive, transitive, ditransitive)
  - sit, like, give
- Two forms of ditransitives
  - give something to somebody, give somebody something
- Singular and plural forms
  - likes, like
- Past participle forms for transitive verbs
  - liked

Content Words: Implementation

- Full-form common lexicon of close to 100'000 entries
- Users can import domain-specific lexicons of content words that possibly override entries of the common lexicon
- Users can temporarily introduce missing content words by prefixing them with their respective word class
  - *A a:trusted man a:deliberately v:backs-up the n:web-page of the n:pizza-delivery-service.*
- The parser can in many cases guess the correct word class of an unknown content word on the basis of its context

ACE Function Words

- Predefined function words
  - determiners, quantifiers, prepositions, coordinators, negation words, pronouns, query words, copula *be,* Saxon genitive marker *'s*
  - natural numbers
- Predefined fixed phrases
  - *there is/are … such that*
  - *it is false that …*
- The set of function words is limited and unchanging
- Not all English function words are in ACE
  - e.g. hence, whom

Construction Rules: Noun Phrases

- Singular countable noun phrases: *a/the/1 card*, *no card*, *every/each card*, *not every/each card, for every/each card*
- Plural countable noun phrases: *the cards*, *some cards*, *3 cards*
- mass noun phrases: *some water*, *no water*, *all water*, *not all water*, *for all water*
- Proper names: *John*, *Mr-Miller*
- (Non-) reflexive (possessive) pronouns: *he/she/it/they, him/her/it/them, himself/herself/itself/themselves, his/her/its/their, his/her/its/their own*
- Indefinite pronouns: *someone*, *somebody*, *something*, *no one*, *nobody*, *nothing*, (*not*) *everyone*, (*not*) *everybody*, (*not*) *everything*
- Generalised quantifiers: *at least 2 cards*, *at most two cards*, *more than 10 cards*, *less than 3 cards*
- Measurement noun phrases: *2 kg of apples, 3 cubicmeter of water*

Construction Rules: Plural Noun Phrases

- ACE plural noun phrases have a collective or a distributive reading.
- The collective reading is the default

  A clerk enters 2 cards.
- The distributive reading is indicated by *each of*

  A clerk enters each of 2 cards.
- NP conjunction gives a plural object

  (each of) a customer and a clerk

Construction Rules: Modifying Noun Phrases

- Adjective: *a rich customer, some cold water*
- Adjective conjunction: *a rich and famous customer*
- *of*-prepositional phrase: *a customer of John*
- Saxon genitive: *John's customer*
- Possessive pronoun: *his (own) card*
- Quoted string as apposition: *a message "John's fault"*
- Variable as apposition: *a customer X*

  (NB: Variables introduced as appositions can be used anaphorically as noun phrases, e.g. *A customer X waits*. *X is tired*.)

Modifying Noun Phrases: Relative Clauses

- Relative clause: *a customer who knows John*
- Relative clause (with inversion): *a customer who John knows*
- Complex relative clauses
  - conjunction: *a customer who is rich and who is famous*
  - disjunction: *a customer who is rich or who is famous*
  - embedding: a customer who sees a man who knows John
  - embedding (with inversion): a customer who a man who knows John sees
- Although syntactically possible, users are advised not to use the relative clause inversion as this results in sentences that are difficult to read.

Construction Rules: Verb Phrases

- Intransitive (*wait*), transitive (*enter something, wait-for something*), and ditransitive verbs (*give something to somebody, give somebody something*)
- 3rd person singular/plural, present tense, active/passive
- Modality (*can, must*)
- Intentionality (*believe that*)
- Prepositions of prepositional verbs and phrasal particles of phrasal verbs must be hyphenated to the verb (*wait-on*, *look-up*, *apply-for*)

Construction Rules: Verb Phrases

- Copula *is*/*are* plus
  - noun phrase: *John is a rich customer*.
  - adjective: *John's wealth is enormous*.
  - comparative adjective: *John is richer than Mary*.
  - transitive adjective: *John is interested-in Mary and fond-of Bill*.
  - prepositional phrase: *John is in his own office*.

Construction Rules: Modifying Verb Phrases

- Adverbs follow the verb or – if present – its complements
  - *A customer waits patiently.*
  - *A customer inserts a card manually.*
- Adverbs can also precede the verb
  - *A customer manually inserts a card.*
- Adverbs can be conjoined (but not disjoined)
  - *A customer inserts a card carefully and manually.*
- Prepositional phrases can be conjoined (but not disjoined)
  - *A customer inserts a card in the bank at a time T.*
- Adverbs and prepositional phrases can be conjoined (but not disjoined)
  - *A customer inserts a card carefully into the slot.*
  - *A customer carefully inserts a card into the slot.*

Construction Rules: Complement vs Adjunct

- Notice the difference between a prepositional/phrasal verb and a verb with a prepositional phrase

  A steward waits-on a table.

  vs.

  The food waits on the table.

  A student is interested-in a course.

  vs.

  A student is interested in a classroom.

Construction Rules: Verb Phrase Coordination

- VPs can be coordinated by *and* and *or*
- Conjunction
  - *A screen flashes and blinks.*
- Disjunction
  - *A screen flashes or blinks.*
- Combinations of conjunctions and disjunctions follow standard binding order of conjunction and disjunction
  - *A screen {flashes and blinks} or is dark.*
- the order can be overridden by commas
  - *A screen flashes, and {blinks or is dark}.*
- NB: The brackets {} are not part of ACE and are only used here to make explicit the binding order!

Construction Rules: ACE Texts

- An ACE text is a sequence of anaphorically interrelated declarative sentences optionally followed by one interrogative sentence.
- Declarative sentences
  - end with full stop
  - can be simple or composite
- Interrogative sentences
  - end with a question mark
  - query the contents of ACE texts

Construction Rules: Simple Sentences

- Simple sentences have the structure
  - subject + predicate + complements + adjuncts
- Complements are the direct and indirect objects
- Adjuncts are optional adverbs and prepositional phrases
- Examples
  - *A customer waits*.
  - *A customer inserts a card*.
  - *A customer gives a card to a clerk*. (alternatively: *A customer gives a clerk a card*.)
  - *A customer inserts a card manually into a slot*.

Construction Rules: *there is* Sentences

- It is possible to create well-formed simple sentences without a verb by using the *there is/are* construct that introduces only an object
  - *There is a customer.*
- No adjuncts or complements are allowed (because there is no main verb)
  - **There is a customer in the bank.*
  - NB: the star (*) means here that the sentence is syntactically incorrect
- Relative clause is possible (because the noun is present)
  - *There is a customer who waits.*

Construction Rules: Composite Sentences

- Composite sentences are recursively built from simpler sentences with the help of the predefined constructors
  - coordination
  - quantification
  - negation
  - subordination
- Example
  - *If a customer inserts a card that is valid then the automatic teller accepts the card and displays a message.*

Construction Rules: Coordination

- Sentences can be coordinated by *and* and *or*
- Sentence conjunction
  - *The screen blinks and John waits.*
- Sentence disjunction
  - *The screen blinks or John waits.*
- Overriding of standard binding order by commas
  - *The screen blinks or John waits, and Mary sleeps.*

Construction Rules: Quantification

- Existential quantification
  - *There is a card. There is some water.*
  - *John enters a card. John drinks some water.*
- Universal quantification
  - *John enters every card.*
- Global existential quantification
  - *There is a code that every clerk enters.*
  - or equivalently: *There is a code such that every clerk enters it.*
  - or equivalently: *There is a code that is entered by every clerk.*
- Global universal quantification
  - *For every code (there is) a clerk (such that he) enters it.*

Construction Rules: Negation

- NP negation
  - *John enters no code.*
- Negated universal quantifier
  - *John enters not every code.*
- Negated generalised quantifiers
  - *John enters not more than 2 cards.*
- VP negation
  - *John does not enter a code.*
- Negated copula
  - *Some water is not drinkable.*
- Sentence negation
  - *It is false that a screen blinks.*
  - *It is false that a screen blinks and that the computer sleeps.*

Construction Rules: Subordination

- ACE knows several forms of subordination
  - relative phrases (we discussed it already when talking about noun phrase modification)
  - conditional sentences
  - sentence subordination
  - modality

Construction Rules: Conditional Sentences

- Conditional sentences are built with the help of *if … then*
  - *If John enters a card then the automatic teller accepts it.*
- Equivalence of universally quantified and conditional sentences

  Every customer enters a card.

  is equivalent to

  If there is a customer then the customer enters a card.

Construction Rules: Sentence Subordination

- New feature of ACE 5
- Negation
  - It is false that a customer inserts a card.
- Negation as failure (to support translation of ACE into languages like Prolog)
  - It is not provable that a customer inserts a card.
- Sentence as an object of a verb
  - A clerk believes that a customer inserts a card.

Construction Rules: Modality

- New feature of ACE 5
- Possibility
  - A trusted customer can insert a card.
  - It is possible that a trusted customer inserts a card.
- Necessity
  - A trusted customer must insert a card.
  - It is necessary that a trusted customer inserts a card.

Construction Rules: Interrogative Sentences

- ACE allows two forms of interrogative sentence
  - yes/no queries
  - wh-queries
- Yes/no queries
  - *Does John enter a card?*
  - *Is the card valid?*
- Wh-queries
  - *Who enters what?*
  - *Which customer enters a card?*
  - *How does John enter a card?*

Constraining Ambiguity: Structural Ambiguity

- To constrain the ubiquitous structural ambiguity of natural language ACE employs three simple means
  - some ambiguous constructs are not part of ACE; unambiguous alternatives are available in their place
  - all remaining ambiguous constructs are interpreted deterministically on the basis of a small set of *interpretation rules*
  - users can accept the assigned interpretation, or they must rephrase the input to obtain another one
- ACE sentences are not ambiguous; however the same sentences can be ambiguous in full English

Interpretation Rules: Ambiguity

- Prepositional phrases modify the verb not the noun
  - *A customer {enters a card with a code}.*
- Relative clauses modify the immediately preceding noun
  - *A customer enters {a card that carries a code} and opens an account.*
- To express coordination within the relative clause the relative pronoun has to be repeated
  - *A customer inserts {a card that is valid and that has a code}.*

Interpretation Rules: Ambiguity

- The scope of sentence negation *it is false that* extends to the end of a simple sentence

  {It is false that a man waits} and a dog barks.
- To express coordination within the scope of sentence negation the word *that* has to be repeated

  {It is false that a man waits and that a dog barks}.
- In *if-then*-sentences the scope of the *if*-part and the scope of the *then*-part extend to the end of a coordination

  {If a man waits and a dog barks} then {a woman smiles and a cat sleeps}.

Interpretation Rules: Ambiguity

- If an adverb can modify the preceding or the following verb then it refers to the preceding verb
  - *A customer who {enters a card manually} types a code.*
- The textual position of a quantifier opens its scope that extends to the end of the sentence, or in a coordination to the end of the respective coordinated phrase
  - *A customer types every code.* ∃∀
  - *Every customer types a code.* ∀∃

Constraining Ambiguity: Plural Noun Phrases

- Plural NPs are highly ambiguous
- Of the many readings of plural NPs ACE provides only the collective and the distribute readings
- Collective reading is the default
  - *A clerk enters 2 cards.*
- Distributive reading is indicated by *each of*
  - *A clerk enters each of 2 cards.*

Constraining Ambiguity: Lexical Ambiguity

- In general, controlling lexical ambiguity is the responsibility of the users.
- Some nouns occur both as countable and as mass; this lexical ambiguity is easily resolved by the correct use of determiners
  - countable noun: *a/the/every laundry*, *the laundries*, *some laundries, …*
  - mass noun: *some laundry*, *all laundry*, …
- Further lexical ambiguity between countable and mass nouns is resolved by the parser preferring the countable interpretation; the mass interpretation can be enforced using the *of*-propositional phrase
  - countable: *John's laundry* mass: *some laundry of John*
  - countable: *her laundry* mass: *some laundry of her*
  - countable: *no laundry* mass: *some laundry does/is not, it is false that some laundry*

Constraining Ambiguity: Lexical Ambiguity

- Verbs are highly ambiguous, since the same verb can appear in the categories intransitive, transitive and ditransitive, and furthermore can occur with and without phrasal particles and prepositions as integral constituents.
- To constrain this type of lexical ambiguity ACE expects that the phrasal particle of a phrasal verb (look up, drop out, shut down) and the preposition of a prepositional verb (look at, apply for) are hyphenated to the verb
  - *A steward waits-on the table.* (vs. *The food waits on the table*.)
  - *John looks-up an entry.* (vs. *John looks up the alley*.)
  - *What does John apply-for?* (vs. *John applies for the second time*.)

Constraining Ambiguity: Lexical Ambiguity

- Hyphenation does not apply to ditransitive verbs since the prepositional complement is not adjacent to the verb and does not easily lead to ambiguity
  - *John gives a card to a clerk.*
  - *Who does John give a card to?*
- Hyphenation can lead to ACE constructs not acceptable in full English
  - *There is an entry. John looks-up it.*

  that can easily be avoided using a definite noun phrase or a variable instead of a pronoun to express the anaphoric reference
  - *There is an entry. John looks-up the entry.*
  - *There is an entry E. John looks-up (the entry) E.*

Evaluation of Disambiguation

- Advantages of constructive disambiguation
  - automatic and efficient disambiguation
  - no use of contextual knowledge, domain knowledge, ontologies
  - simple, systematic, general, easy to learn interpretation rules
  - reliable, reproducible and thus intelligible behaviour
- Open problems
  - rules do not always lead to natural interpretation
  - sometimes result in stilted English
  - Can we control all ambiguities with this strategy?
  - Does strategy scale up to larger fragment of ACE?

Anaphoric References

- ACE texts are interrelated by anaphoric references, i.e. references to textually preceding noun phrases
- Anaphoric references can be made by
  - proper names: *John*
  - pronouns: *it*, *itself*
  - definite noun phrases: *the card, the water, the red card, the man who waits*
  - variables: *the card X*, *X*
- E.g. *John has a customer. John inserts his card and types a code X. Bill sees X. He inserts his own card and types the code.*

Interpretation Rules: Anaphoric References

- Proper names like *John* or *Mr-Miller* always denote the same object and thus serve as their own anaphoric references
- In all other cases resolution of anaphoric references is governed by
  - accessibility
  - recency
  - specificity
  - reflexivity

Interpretation Rules: Accessibility

- A noun phrase is not accessible if it occurs in a negated sentence
  - *John does not enter a card. *It is correct.*
- A noun phrase is not accessible if it occurs in a conditional sentence
  - *Every customer has a card. *It is correct.* (use instead: *Every customer has a card that is correct.*)
- **but** a noun phrase in the *if*-part of a conditional sentence is accessible in the *then*-part
  - *If a customer has a card then he enters it.*
- A noun phrase in a disjunction is only accessible in subsequent disjuncts
  - *A customer enters a card or drops it. *It is dirty.*

Interpretation Rules: Pronominal References

- If the anaphor is a non-reflexive personal pronoun (*he*, *him*, …) or a non-reflexive possessive pronoun (*his*, …) then the anaphor is resolved with the most recent accessible noun phrase that agrees in gender and number, and that is not the subject of the sentence.
- Examples
  - *John has a card. Bob sees him and takes it.*
  - **John sees his wife.* (use: *John sees his own wife.*)

Interpretation Rules: Pronominal References

- If the anaphor is a reflexive personal pronoun (*herself*, …) or a reflexive possessive pronoun (*her own*, …) then the anaphor is resolved with the subject of the sentence in which the anaphor occurs if the subject agrees in gender and number with the anaphor.
- Example
  - Mary takes her own card and gets some money for herself.

Interpretation Rules: Definite Noun Phrases

- If the anaphor is a definite NP then it is resolved with the most recent and most specific accessible noun phrase that agrees in gender and number.
- Example
  - *There is a blue ball. There is a red ball. John sees the ball. Mary sees the blue ball.*
- If a definite NP cannot be resolved then it is interpreted as an indefinite noun phrase introducing a new object.

Interpretation Rules: Variables

- If the anaphor is a variable then it is resolved with an accessible noun phrase that has the variable as apposition.
- Example
  - *John has a card X and a card Y. Mary takes the card. Bob takes the card X. Harry takes Y.*
- Example: predecessor is not accessible
  - *If a customer has a card C then the customer enters C. *C is not valid.*

Using ACE

- One can talk ACE to one's friends and relatives but the main idea is communicate with the computers
- Learning ACE: reading the documentation, playing with the parser and other tools (paraphraser, reasoner, various converters)
- Using ACE: as a specification, ontology, and rule language; as a natural language front-end, for verbalization; in teaching formal languages

Relevant Documentation

- ACE in a Nutshell is a short overview of the ACE language.
- ACE Lexicon Specification describes the allowed content words
- ACE Construction Rules lists the rules that determine which sentences belong to ACE.
- ACE Interpretation Rules lists the rules that remove the ambiguity from the ACE sentences.
- ACE Styleguide describes how to use ACE, including how to avoid pitfalls.

Very Brief Style Guide

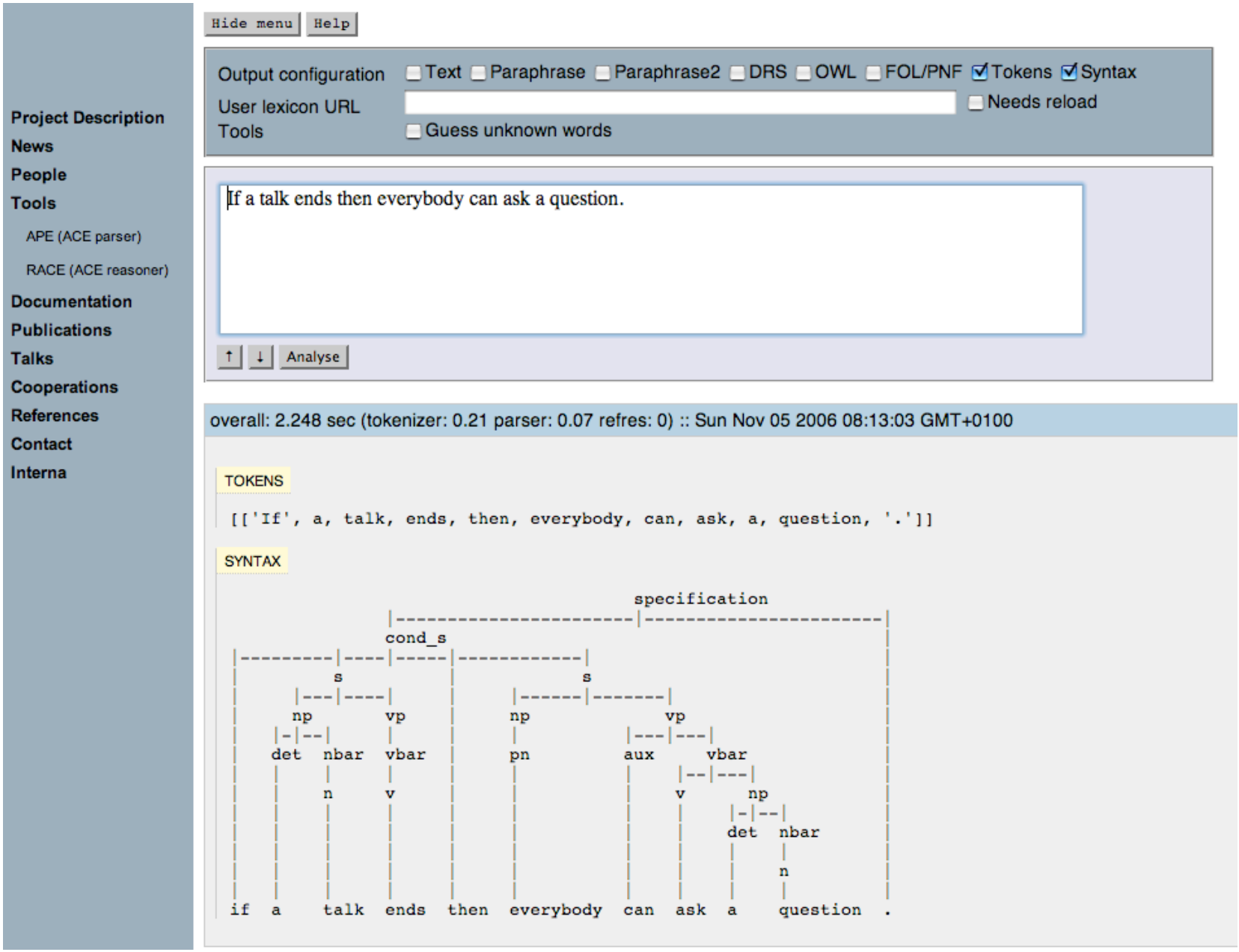
- While the ACE parser will unravel any syntactically correct sentence, however complex, *you* may have problems to do so.
- Remember that your text is the only source of information; there is no hidden knowledge.
- Concrete proposals
  - avoid unnecessarily complex constructs involving coordination, quantification and negation
  - instead of a complex sentence, use several simpler sentences ones, connecting them by anaphoric references

Tool Support

- ACE parser (APE)
- ACE paraphraser (DRACE)
- ACE reasoner (RACE)
- Various converters
- Graphical user interfaces to those tools

ACE Parser

- Detects syntactic errors and unknown words in an ACE text
- Pinpoints those errors to the user so that they can be quickly removed
- Outputs the analysis of the text
  - tokenization
  - syntax trees
  - discourse representation structure

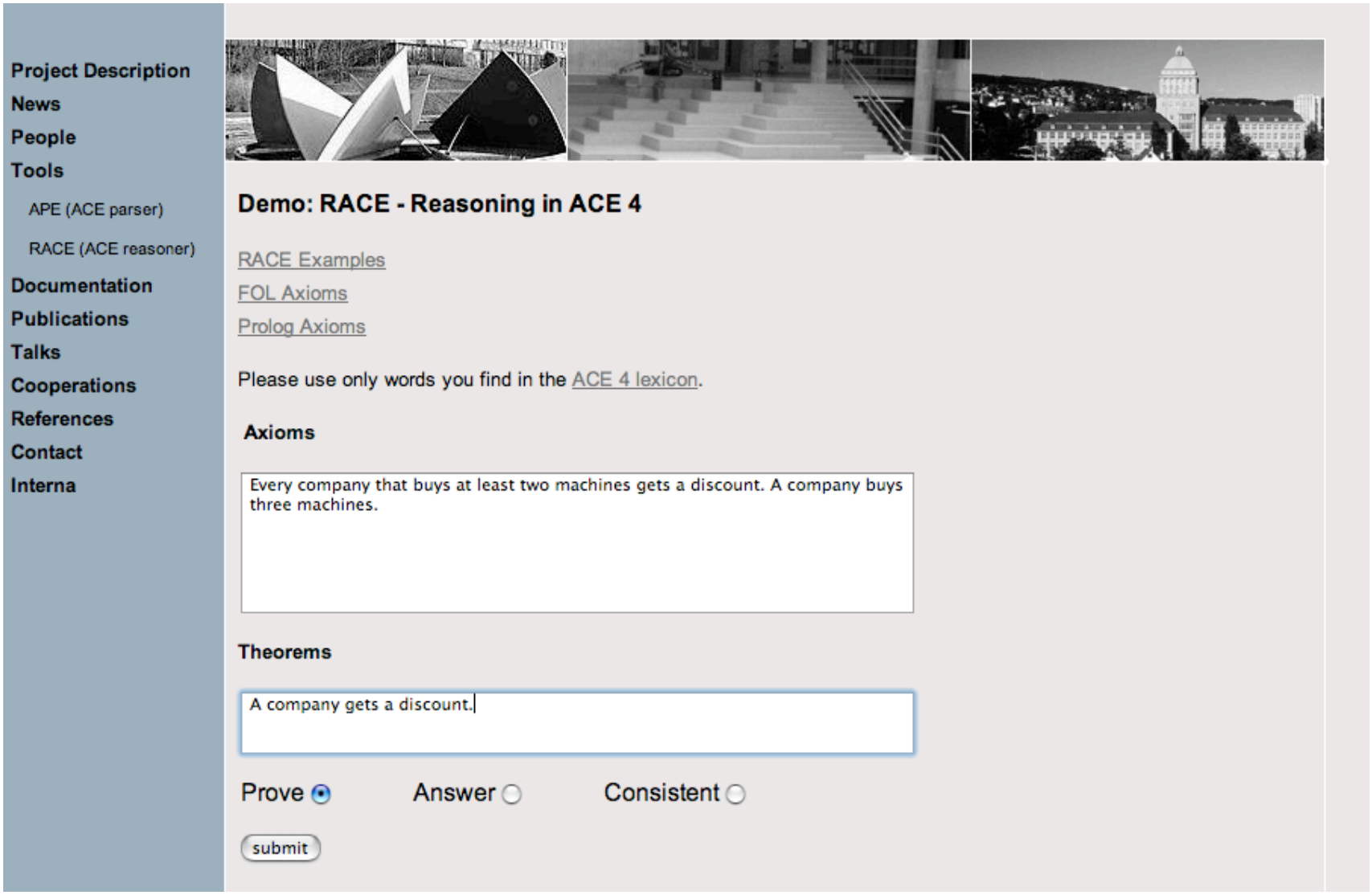

Project Description
News
People
Tools
APE (ACE parser)
RACE (ACE reasoner)
Documentation
Publications
Talks
Cooperations
References
Contact
Interna
Hide menu
Help
Output configuration
Text
Paraphrase
Paraphrase2
DRS
OWL
FOL/PNF
Tokens
Syntax
User lexicon URL
Needs reload
Tools
Guess unknown words
If a talk ends then everybody can ask a question.
↑
↓
Analyse
overall: 2.248 sec (tokenizer: 0.21 parser: 0.07 refres: 0) :: Sun Nov 05 2006 08:13:03 GMT+0100
TOKENS
[['If', a, talk, ends, then, everybody, can, ask, a, question, '.']]
SYNTAX
specification
cond_s
s
s
np
vp
np
vp
det
nbar
vbar
pn
aux
vbar
n
v
v
np
det
nbar
n
if
a
talk
ends
then
everybody
can
ask
a
question
.

ACE Paraphraser

- Converts an ACE text into a semantically equivalent but syntactically different ACE text
- E.g.
  - Input: John likes everybody.
  - Paraphrase 1: If there is somebody X then John likes X.
  - Paraphrase 2: Everybody is liked by John.
- This helps the user to understand how his/her text is interpreted in ACE

ACE Reasoner

- Detects inconsistencies in the ACE text, pinpointing the minimal inconsistent subset(s) of the text
- Detects if a given ACE text is redundant (i.e. doesn't provide any new information) with regards to another ACE text
- Answers questions about the text
  - Who does John like?


Project Description
News
People
Tools
APE (ACE parser)
RACE (ACE reasoner)
Documentation
Publications
Talks
Cooperations
References
Contact
Interna
Demo: RACE - Reasoning in ACE 4
RACE Examples
FOL Axioms
Prolog Axioms
Please use only words you find in the ACE 4 lexicon.
Axioms
Every company that buys at least two machines gets a discount. A company buys three machines.
Theorems
A company gets a discount.
Prove
Answer
Consistent
submit

Project Description
News
People
Tools
APE (ACE parser)
RACE (ACE reasoner)
Documentation
Publications
Talks
Cooperations
References
Contact
Interna

RACE Results

New Proof

Runtime 9970 milliseconds

RACE proved that the sentence(s)
A company gets a discount.
can be deduced from the sentence(s)
Every company that buys at least two machines gets a discount.
A company buys three machines.

using the FOL axiom(s)

Number Axiom 1202.

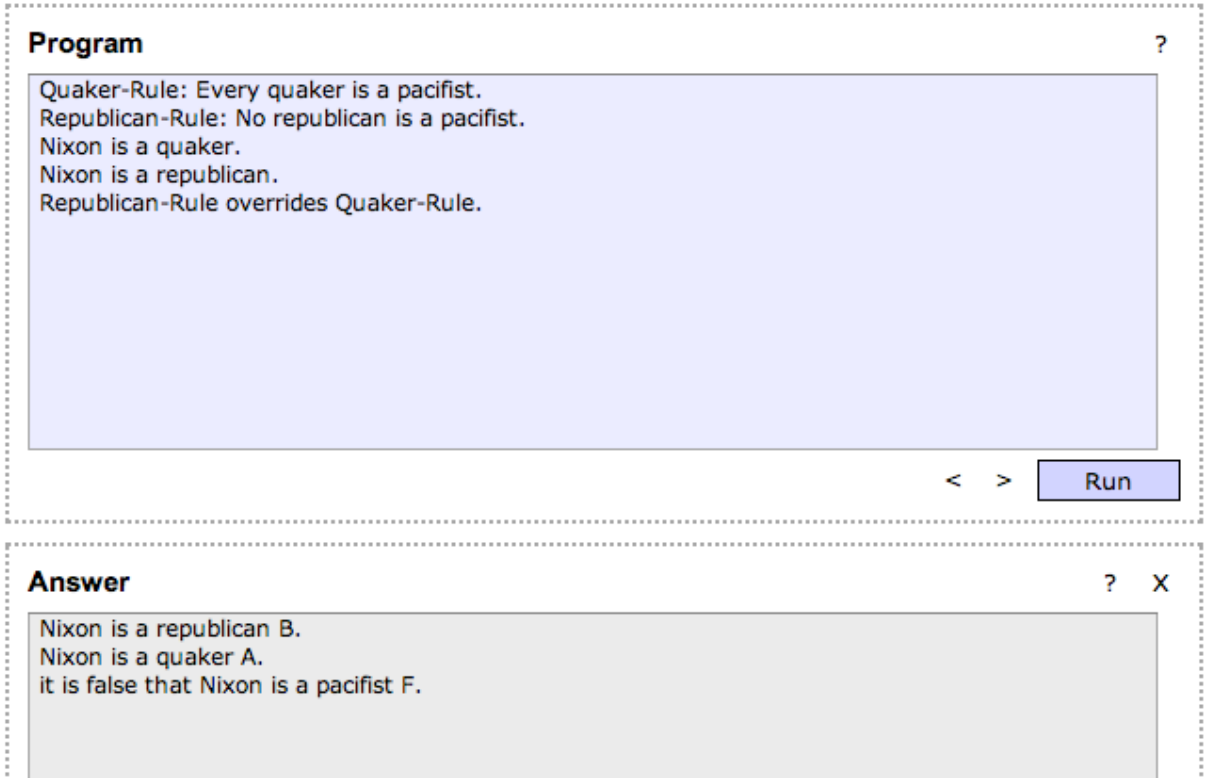
FOL Axioms

Converters

- Why convert ACE to another formal language?
  - other languages might provide solid tool support (editors, reasoners, visualizers, storing, sharing)
- Why convert other languages to ACE?
  - access to existing (large) knowledge bases
  - understand existing knowledge bases better (by the more natural representation that ACE provides)

Existing Converters

- Converting ACE to standard representation of first-order logic, prenex normal form, various clausal forms
- Converting ACE to rule languages, e.g. Courteous Logic Programs (extension of Prolog), RuleML
- Converting ACE to OWL (Semantic Web Ontology Language)
- Converting OWL to ACE


Program
?
Quaker-Rule: Every quaker is a pacifist.
Republican-Rule: No republican is a pacifist.
Nixon is a quaker.
Nixon is a republican.
Republican-Rule overrides Quaker-Rule.
< >
Run
Answer
? X
Nixon is a republican B.
Nixon is a quaker A.
it is false that Nixon is a pacifist F.