Tutorial Dialogue in DiBEx

Manfred Klenner

Computational Linguistics

University of Zürich

Switzerland

Tutorial Dialogue for CALL?

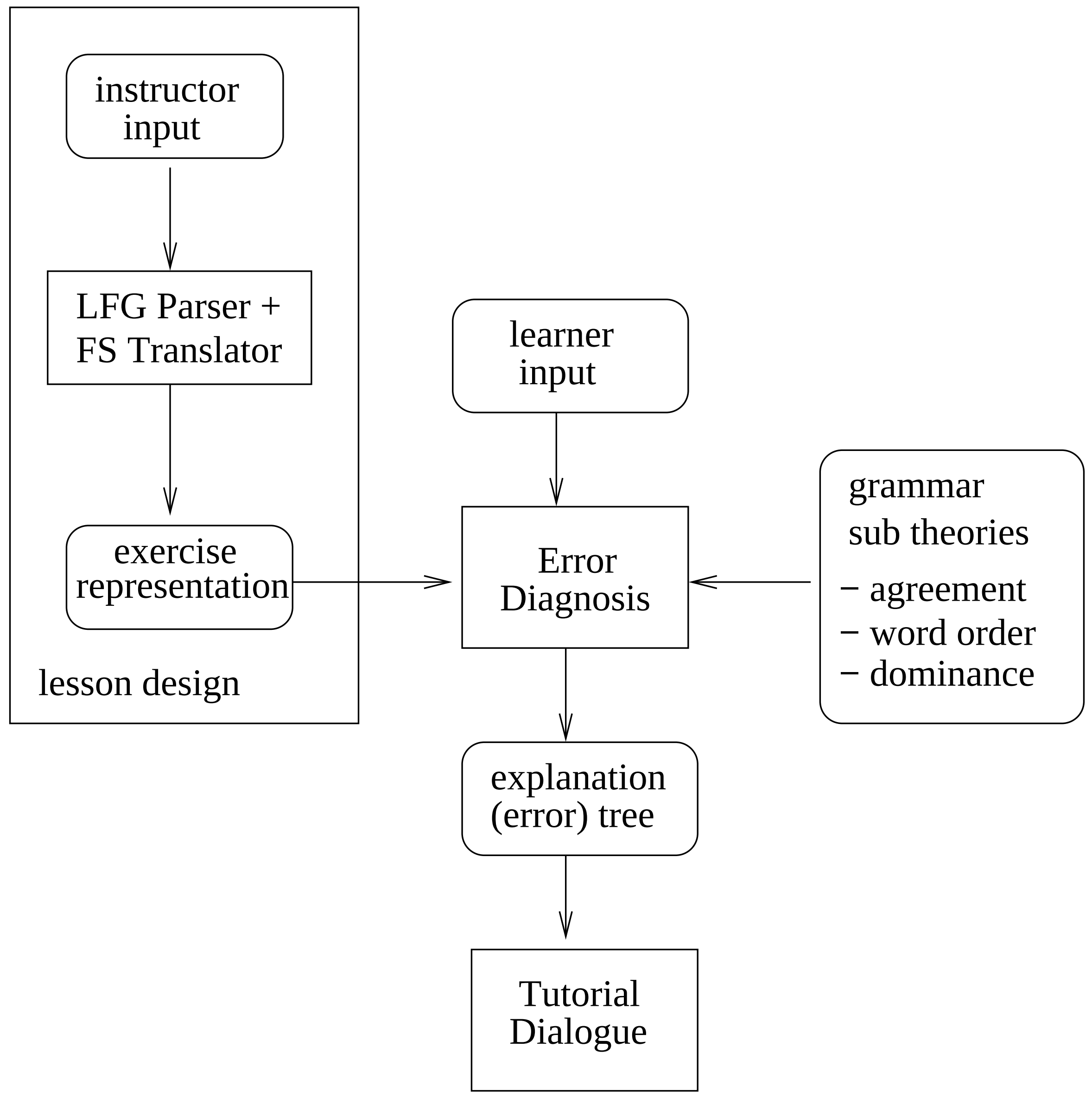
- YES!
- what if the student does not understand / cannot utilize
  - a canned error message
  - or an otherwise generated (more sophisticated) error feedback
  - ⇒ and would like to have an explanation ...?
- also:
  - second language learning is a kind of problem solving
  - cognitive psychologists claim:

    “self explanations enhance problem solving skills”
  - tutorial dialogue can be tailored to force self explanations (of the mistakes being made)
- and of course: tutorial dialogue is an interesting research problem ...

The DiBEx Tutor

- DiBEx is a dialogue-based explanation facility for CALL systems with the following features:
  - simple learning tasks: translate/form a sentence
  - correct solution is provided to DiBEx by a lesson instructor
  - error diagnosis simplified:
    = comparison of the correct solution to the student input
  - grammatical knowledge is represented explicitly by various subtheories (agreement, word order, ...)
  - subtheories are operational:
    = student input passes a ’subtheory call’ or not
  - if a subtheory call fails, a meta interpreter generates an error explanation tree
  - explanations are based on a ’Socratic’ tutorial strategy
- note: DiBEx is a *prototype* (Prolog implementation)
- current focus: grammar representation as horn clause theories

DiBEx: System Architecture



The Learning Task

- translating a sentence, or
- forming a sentence given a couple of words
- example:

given:	’Mond’, ’aufgehen’ (’moon’, ’rise’)
task:	form a full-inflected sentence (present perfect)
solution:	’Der Mond ist aufgegangen’ (’the moon has risen’)

- but what if:

’Die Mond ist aufgegangen’

- error diagnosis / correction advise:

the article is wrong / replace ’die’ with ’der’

- but there are a lot of alternative ways to react ....

Error Feedback

user: ’Die Mond ist aufgegangen’
system:

1. ’die’ is wrong (is feminine)
2. the wrong article was selected
3. ’Mond’ is masculine
4. the noun is masculine
5. the noun is masculine, but the article is not
6. there is an agreement error
7. there is an agreement error within the subject
8. ’die’ and ’Mond’ do not agree
9. the article and noun do not agree
10. ’die’ and ’Mond’ do not agree in a grammatical category
11. ’die’ and ’Mond’ do not have the same gender
12. a grammatical category of the subject is wrong
13. the subject is wrong

Sample Dialogue

user	’die Mond ist aufgegangen’
	(the moon has risen)
system:	there is an agreement error
user:	where?
system:	at the subject
user:	why?
system:	the article and the noun do not agree in gender
user:	I can’t fix the problem
system:	German ’Mond’ is masculine
user:	I see, but what am I supposed to do?
system:	choose a masculine article
user:	what is it?
system:	’der’

Representing Grammar Theory for Error Explanation

- grammar theory as a source of error explanation should be
  - represented explicitly (i.e. being explainable)
  - operational (failures can be traced)

- we adopt Menzel's scheme (1992) for the representation of facts

⇒ val(der,case,nominative), ...

- we use horn clause representation of rules (Prolog clauses)

⇒ agreement(Sent) :-
forall(phrase_of_sent(Sent,Phrase),phrase_agreement(Phrase))

What about Contemporary Syntax Formalisms?

- the trouble with contemporary syntax formalisms (e.g. feature based approaches):
  - knowledge about grammar is left implicit (e.g. agreement: realised by structure sharing, verified by unification)
  - e.g. the fact that ’every part of a noun phrase must agree with the head in case, number and gender’ is buried within a complex machinery,
  - and various things are done parallel (e.g. while proving word order, agreement is being checked, etc.)

Grammar Facts

'Der Mond (ist aufgegangen)'	'Die Mond (ist aufgegangen)'
`val(main1,isa,main_clause).`	
`val(subject,has_part,[det1,noun1]).`	
`val(subject,isa,noun_phrase).`	
`val(det1,token,der).`	`val(det1,token,die).`
`val(noun1,token,mond).`	
`val(der,case,nom).`	`val(die,case,nom).`
`val(der,gen,mas).`	`val(die,gen,fem).`
`val(der,num,sg).`	`val(die,num,sg).`
`val(mond,case,nom).`	
`val(mond,gen,mas).`	
`val(mond,num,sg).`	
`...`	

Grammar Theory: Agreement

```
agreement(Sent) :-
     subject_verb_agreement(Sent),
     forall(phrase_of_sent(Sent,Phrase),phrase_agreement(Phrase)).

phrase_agreement(Phrase) :-
     val(Phrase,isa,noun_phrase),
     is_head(Phrase,Head),
     forall(non_head(Phrase,NonHead),
            cng_agreement(NonHead,Head)).

phrase_of_sent(Sent,Part) :- ....
is_head(Phrase,Head) :- .....
non_head(Phrase,NonHead) :- .....

cng_agreement(NonHead,Head) :-
     forall(member(Feature,[case,num,gen]),agree(Head,NonHead,Feature)).

agree(Head,NonHead,Feature) :-
     val(Head,Feature,Val), val(NonHead,Feature,Val).
```

Error Explanation

everything to explain an erroneous student input is at hand:

- grammar subtheories
- student input (a set of ’val facts’)
- correct solution (a set of ’val facts’)
- and a meta interpreter that applies a subtheory to
  - the student input
  - the correct solution

  producing an error explanation tree which is
  - an instantiation of a subtheory (technically: a trace)
  - with at least one branch indicating the violated principles

Error Explanation Tree

error explanation tree:

- top: the violated theory
- bottom: the violated fact(s)
- intermediate: error explanations at different levels of abstraction

child explains mother:

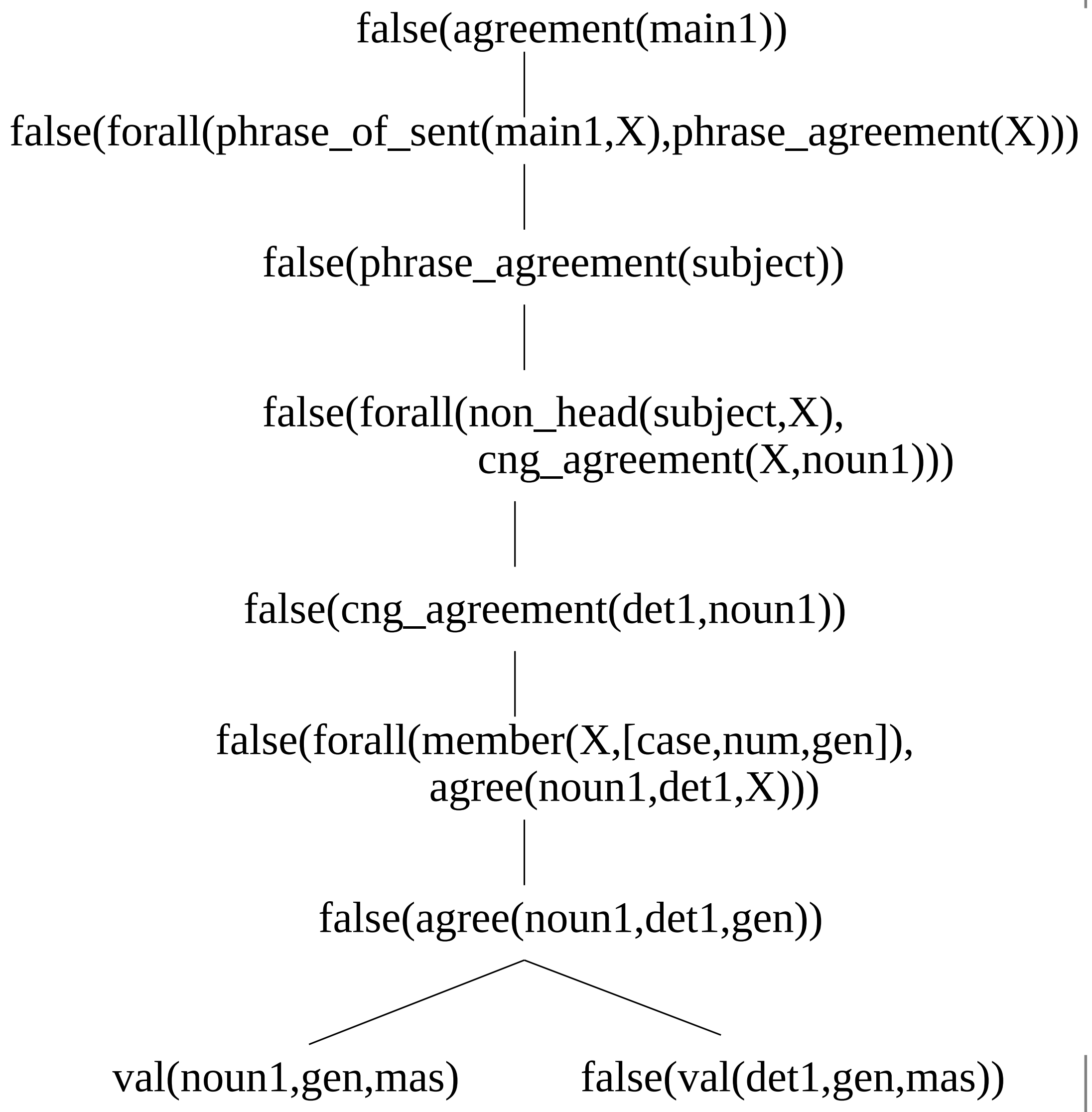
every (child) node is an explanation to the failure of its mother node

for example:

mother:
- det1 and noun1 don't agree in gen(der)

children:
- the gender of noun1 is mas(culine), but
- the (gender) of det1 is not.

why questions: descend the 'false branch' down to an informative node


false(agreement(main1))
false(forall(phrase_of_sent(main1,X),phrase_agreement(X)))
false(phrase_agreement(subject))
false(forall(non_head(subject,X), cng_agreement(X,noun1)))
false(cng_agreement(det1,noun1))
false(forall(member(X,[case,num,gen]), agree(noun1,det1,X)))
false(agree(noun1,det1,gen))
val(noun1,gen,mas)
false(val(det1,gen,mas))

A Socratic Tutorial Strategy

DiBEx currently uses the following tutorial strategy:

- guide the student to identify the faulty fact in a top down manner, i.e. identify the grammatical principle violated by the input
- provide sparse information, i.e. explain everything on an abstract level if possible (e.g. refer to a word by its word class)
- introduce at each explanation step at least one piece of information that is new to the student
- avoid becoming too vague (e.g. don’t use more than one indefinite description, don’t become to abstract, e.g. ’a function word is wrong’)

The Tutor at Work, Part I

user: 'die Mond ist aufgegangen'
(the moon has risen)

system: there is an agreement error

user: where?

system: at the subject

user: why?

system: the article and the noun do not agree in gender

user: I can't fix the problem

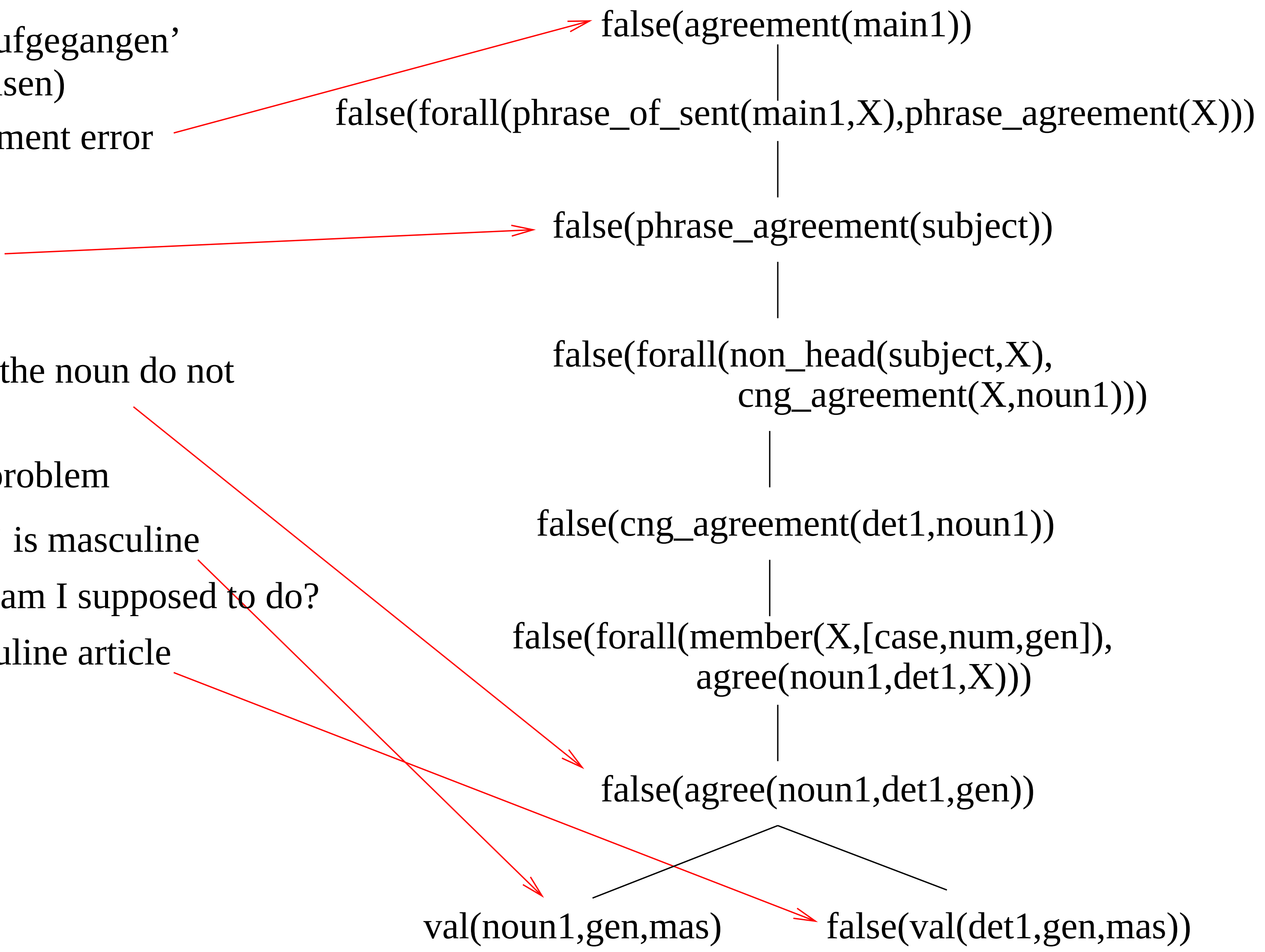
system: German 'Mond' is masculine

user: I see, but what am I supposed to do?

system: choose a masculine article

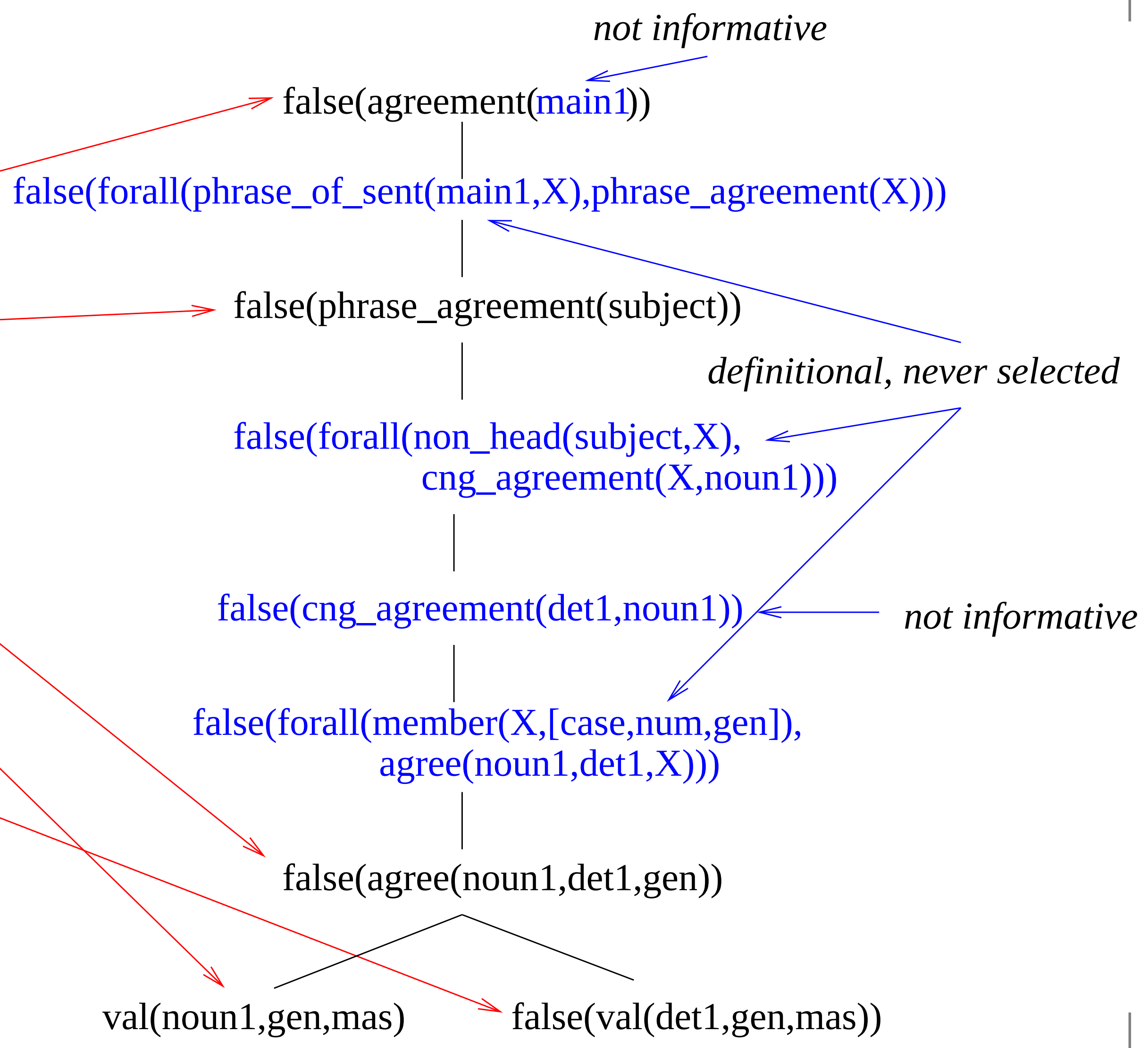
user: what is it ?

system: 'der'


false(agreement(main1))
false(forall(phrase_of_sent(main1,X),phrase_agreement(X)))
false(phrase_agreement(subject))
false(forall(non_head(subject,X), cng_agreement(X,noun1)))
false(cng_agreement(det1,noun1))
false(forall(member(X,[case,num,gen]), agree(noun1,det1,X)))
false(agree(noun1,det1,gen))
val(noun1,gen,mas)
false(val(det1,gen,mas))

The Tutor at Work, Part II

user:	’die Mond ist aufgegangen’ (the moon has risen)
system:	there is an agreement error
user:	where?
system:	at the subject
user:	why?
system:	the article and the noun do not agree in gender
user:	I can’t fix the problem
system:	German ’Mond’ is masculine
user:	I see, but what am I supposed to do?
system:	choose a masculine article
user:	what is it ?
system:	’der’


not informative
false(agreement(main1))
false(forall(phrase_of_sent(main1,X),phrase_agreement(X)))
false(phrase_agreement(subject))
definitional, never selected
false(forall(non_head(subject,X), cng_agreement(X,noun1)))
false(cng_agreement(det1,noun1))
not informative
false(forall(member(X,[case,num,gen]), agree(noun1,det1,X)))
false(agree(noun1,det1,gen))
val(noun1,gen,mas)
false(val(det1,gen,mas))

Filtering

- mutual knowledge: ’agreement error’ at the ’subject’
- false(cng_agreement(det1, noun1))

  ⇒ ’article and noun do not agree in case, number and gender’
- nothing new, since:
  - the definition of ’phrase agreement’ implies ’agreement in *case, number and gender*’
  - det1 and noun1 are the only parts of the subject phrase
- different situation if: ’Die rote Mond ..’ (’the red moon ..’)
  - ’article and noun’ is now opposed to ’adjective and noun’
  - ⇒ ’article and noun do not agree ..‘ is informative

Outlook

- done
  - lesson translator (teacher sentence ⇒ correct solution)
  - error diagnosis (student input ⇒ error explanation tree)
  - a discourse model including an evolving coherence model
  - partially covered: agreement, word order, dominance, valency
  - questions: why (explanation), where etc. (concretion)
- to be done
  - more grammar theory
  - a (heuristic) strategy for error hypotheses selection
  - an (explicit) user model
  - other tutorial strategies
  - a strategy to cope with multiple errors
  - ...