Computing Weakest Readings: Failures and Perspectives

Alexander Koller

EGK Seminar
Saarbrücken
4 July 2002
Overview

• Underspecification

• Eliminating Unsatisfiable Readings

• Weakest Readings

• Graph Rewriting: Problems and Perspectives
Ambiguity and Underspecification

- Processing of ambiguities are one of the big challenges for computational linguistics.

- Problem: Number of readings grows exponentially with number of ambiguities.

- Underspecification: Represent all readings in a single, compact description.

- Try to work with descriptions as long as possible; delay enumeration of readings.

- Here: scope ambiguities (Alshawi & Crouch 92, Reyle 93, Muskens 95, Deemter & Peters 96, Pinkal 96, Egg et al. 98, . . . )
Every student attended a workshop.

\[ \forall x. \text{student}(x) \rightarrow \exists y. \text{workshop}(y) \land \text{attend}(x, y) \]

\[ \exists y. \text{workshop}(y) \land \forall x. \text{student}(x) \rightarrow \text{attend}(x, y) \]
not underspecified: map syntax directly to many semantic readings

underspecified: new intermediate level of descriptions

e.g. CLLS, FOL, DRT, IL, ...

FOL, DRT, IL, ...
Sometimes some readings are unsatisfiable:
*Every boy ate a cookie.*

**Goal:** Strengthen the underspecified description to remove unsatisfiable (and other unwanted) readings.

Underspecification opens up the chance of eliminating unwanted readings without ever seeing them.
Every boy ate a cookie.

Two readings are characterized by $X \triangleleft^* Y$ or $Y \triangleleft^* X$. Reading with $Y \triangleleft^* X$ is inconsistent with world knowledge, so can commit to $X \triangleleft^* Y$. 
Eliminating Unsatisfiable Readings

It can be done using the following algorithm:

1. Pick a node with two incoming dominance edges.
2. Consider the strengthened constraint $\varphi' = \varphi \land X \triangleleft^* Y$.
3. If all readings of $\varphi'$ are unsatisfiable, go back to 1 with $\varphi \land Y \triangleleft^* X$.
4. Otherwise, do the same for $\varphi'' = \varphi \land Y \triangleleft^* X$. Then do the same for the other nodes with two incoming dominance edges.
5. Terminate if none of this was successful.

Main problem: How do we check whether all readings of $\varphi'$ are unsatisfiable?
• First-order entailment $A \models B$ establishes a partial order on the set of readings.

• *Every man loves a woman:* 
  $$\exists \forall \models \forall \exists.$$ 

• Call the maximal elements of this order *weakest readings*.

• All readings of a constraint are unsatisfiable iff all weakest readings are unsatisfiable. This can be checked using a theorem prover.

• Weakest readings are independently interesting: Represent safe information.

• Are there always unique weakest readings?
Unfortunately, even very simple sentences do not have unique weakest readings.

*Every student does not pay attention.*

\[(1) \ \forall x. \text{stud}(x) \rightarrow \neg \text{payatt}(x)\]

\[(2) \ \neg \forall x. \text{stud}(x) \rightarrow \text{payatt}(x)\]

(1) \ntriangledown (2): models that contain no students

(2) \ntriangledown (1): some students pay attention, some don’t

But intuitively, (1) is stronger than (2)!
• Strong NPs such as *every student* presuppose that their restriction is non-empty.

• Define a new entailment relation $\models_p$:

\[
A \models_p B \iff A \cup \text{pre}(A) \models B,
\]

where $\text{pre}(A)$ are the presuppositions of $A$.

• Seems to solve the problem: $(1) \models_p (2)$.

• Weakest Readings Hypothesis: *Every sentence has a unique weakest reading, given an appropriate notion of entailment.*
Every researcher of a company does not see a sample.
(18 readings)
Unfortunately, there are sentences for which it is unclear whether they have a unique weakest reading.

*A researcher of every company does not laugh.*

Sentence has five readings. Two that are minimally strong are:

1. \(\forall y. (\text{comp}(y) \rightarrow \exists x. (\text{res}(x) \land \text{of}(x, y) \land \neg \text{laugh}(x)))\)
   “Every company employs a sad researcher.”

2. \(\neg \exists x. (\text{res}(x) \land \forall y. (\text{comp}(y) \rightarrow \text{of}(x, y)) \land \text{laugh}(x))\)
   \(\equiv \forall x. (\text{res}(x) \land \forall y. (\text{comp}(y) \rightarrow \text{of}(x, y))) \rightarrow \neg \text{laugh}(x)\)
   “There is no researcher who works for every company and laughs.”

(2) \(\models\) (1) if there is a researcher who works for every company. But does the reading really presuppose that?
A researcher of every company does not laugh.

Even worse, if we assume that (2) is stronger than (1), we must also accept that (3) is stronger than (1) because (3) \( \models_p (2) \).

\[
(1) \quad \forall y. (\text{comp}(y) \rightarrow \exists x. (\text{res}(x) \land \text{of}(x,y) \land \neg \text{laugh}(x)))
\]

“Every company employs a sad researcher.”

\[
(3) \quad \neg \forall y. (\text{comp}(y) \rightarrow \exists x. (\text{res}(x) \land \text{of}(x,y) \land \text{laugh}(x)))
\]

“Not every company employs a happy researcher.”, i.e. “There is a company that employs only sad researchers.”

But (1) and (3) are intuitively totally incomparable.
Now What?

- It seems we must abandon the Weakest Reading Hypothesis.

- Call minimally strong readings “weakest readings” from now on.

- Many sentences will still have a unique or only a few weakest readings – typically much fewer than total number of readings.

- For such sentences, we can still save a lot of work by working only with weakest readings.
Weakening by Graph Rewriting

- My initial approach to computing weakest readings: Successive manipulations of the constraint graph so described readings become increasingly weaker.

- Add one dominance edge in each step; this separates the set of readings into two halves.

- Rewriting system is **sound** iff whenever \( G \rightarrow G' \), \( G \) and \( G' \) have the same weakest readings.

- Rewriting system is **complete** iff we always end up with a constraint graph that has only weakest (i.e. pairwise incomparable) readings.

- Compute the set of weakest readings by applying the rewriting rule to exhaustion, then solving the last constraint.
“Whenever there is a local scope ambiguity between an indefinite and the scope of a universal quantifier, give the universal wide scope.”

Similar rules can be found for other combinations of $\exists$ and $\forall$. 
Naive Graph Rewriting Doesn’t Work

Unfortunately, this doesn't work even for rather simple (artificial) graphs:

Weakest reading if $X \triangleleft^* Y$: $\forall y. P(y) \rightarrow \exists x. (R(x) \land Q(x, y))$

Weakest reading if $Y \triangleleft^* X$: $\exists x \forall y. ((P(y) \land R(x)) \rightarrow Q(x, y))$

The two readings are incomparable; so there is no sound and complete graph rewriting system that works for this constraint.
• There is no graph rewriting algorithm that is sound and complete for all constraints in general.

• But maybe there is a restricted fragment of all constraints for which such an algorithm can be found!

• Have tried various fragments that are still too big for algorithms.

• Have found various fragments that are too small to be interesting.
- Constraints which are chains whose upper fragments are ordinary first-order quantifiers as they occur in natural language. (No artificial formulas.)

- No negations for now.

- Obvious soundness proof fails, but result may still be true.

- Need to generalize chains.

- Can define a nontrivial grammar that only generates chains.
The Next Steps

• Quest for a useful fragment that allows graph rewriting.

• Maybe graph rewriting is the wrong approach. Could also pick an arbitrary reading and weaken it successively.

• Weakening an arbitrary reading might lead to a formula that is not a reading, but still entailed by all real readings. But this might still be useful.

• Read up on indefinites and presuppositions.
Conclusion

- Want to compute weakest readings so I can remove whole classes of unsatisfiable readings in one step.

- There are sentences that don’t have a unique weakest reading. Existential presuppositions of strong NPs help sometimes, but not always.

- Have explored graph rewriting to compute weakest readings.

- This doesn’t work in general.

- Can I find a useful fragment of the general case for which it does?