Logic-Based Natural Language Processing

Prof. Dr. Michael Kohlhase

Knowledge Representation and -Processing
Computer Science, FAU Erlangen-Nürnberg
`Michael.Kohlhase@FAU.de`

2025-02-06

Elevator Pitch for LBS

- **Mission:** In this course we will
  - explore how to model the *meaning of natural language* via transformation into *logical systems*
  - use *logical inference* there to unravel the missing pieces; the information that is *not linguistically realized*, but is conveyed anyways.

Elevator Pitch for LBS

- **Mission:** In this course we will
  - explore how to model the *meaning of natural language* via transformation into *logical systems*
  - use *logical inference* there to unravel the missing pieces; the information that is *not linguistically realized*, but is conveyed anyways.
- **Warning:** This course is only for you if you like logic!

  You are going to get lots of it and we are going to introduce our own logics, usually a new facet every week or fortnight.

Elevator Pitch for LBS

- **Mission:** In this course we will
  - explore how to model the *meaning of natural language* via transformation into *logical systems*
  - use *logical inference* there to unravel the missing pieces; the information that is *not linguistically realized*, but is conveyed anyways.
- **Warning:** This course is only for you if you like logic!
  You are going to get lots of it and we are going to introduce our own logics, usually a new facet every week or fortnight.
- **Theory in this course:** We wild do so in an abstract, mathematical fashion, but concrete enough that we could implement all moving parts – NL grammars, semantics construction, and inference systems – in meta-grammatical/logical systems.

Elevator Pitch for LBS

- **Mission:** In this course we will
  - explore how to model the *meaning of natural language* via transformation into *logical systems*
  - use *logical inference* there to unravel the missing pieces; the information that is *not linguistically realized*, but is conveyed anyways.
- **Warning:** This course is only for you if you like logic!
  You are going to get lots of it and we are going to introduce our own logics, usually a new facet every week or fortnight.
- **Theory in this course:** We wild do so in an abstract, mathematical fashion, but concrete enough that we could implement all moving parts – NL grammars, semantics construction, and inference systems – in meta-grammatical/logical systems.
- **Practice in PSNLP Project:** We will implement them in the meta-grammatical/logical GLIF system (based on GF, MMT, and ELPI) in the Symbolic NLP Project (5 ECTS; lab work). (see me if you are interested)

Chapter 1
Preliminaries

1.1 Administrative Ground Rules

Prerequisites for LBS

- **Content Prerequisites:** The mandatory courses in CS@FAU; Sem 1-4, in particular:
  - course “Grundlagen der Logik in der Informatik” (GLOIN)
  - some of the CS Math courses “Mathematik C1-4” (IngMath1-4) (math tolerance)
  - algorithms and data structures (programming/complexity)
  - AI-1 (“Artificial Intelligence I”) (for the logic part)

Prerequisites for LBS

- **Content Prerequisites:** The mandatory courses in CS@FAU; Sem 1-4, in particular:
  - course "Grundlagen der Logik in der Informatik" (GLOIN)
  - some of the CS Math courses "Mathematik C1-4" (IngMath1-4) (math tolerance)
  - algorithms and data structures (programming/complexity)
  - AI-1 ("Artificial Intelligence I") (for the logic part)
- **Intuition:** (take them with a kilo of salt)
  - This is what I assume you know! (I have to assume something)
  - In many cases, the dependency of LBS on these is partial and "in spirit".
  - If you have not taken these courses (or do not remember),
    - read up on them as needed! (preferred, do it in a group)
    - We can cover them in class (if there are more of you)

Prerequisites for LBS

- **Content Prerequisites:** The mandatory courses in CS@FAU; Sem 1-4, in particular:
  - course "Grundlagen der Logik in der Informatik" (GLOIN)
  - some of the CS Math courses "Mathematik C1-4" (IngMath1-4) (math tolerance)
  - algorithms and data structures (programming/complexity)
  - AI-1 ("Artificial Intelligence I") (for the logic part)
- **Intuition:** (take them with a kilo of salt)
  - This is what I assume you know! (I have to assume something)
  - In many cases, the dependency of LBS on these is partial and "in spirit".
  - If you have not taken these courses (or do not remember),
    - read up on them as needed! (preferred, do it in a group)
    - We can cover them in class (if there are more of you)
- **The real Prerequisite:** Motivation, interest, curiosity, hard work. (LBS is non-trivial)
- You can do this course if you want! (We will help you)

Assessment, Grades

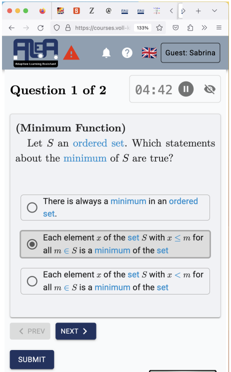
- **Overall (Module) Grade:**
  - Grade via the exam (Klausur) $\leadsto$ 100% of the grade.
  - Up to 10% bonus on-top for an exam with $\geq 50\%$ points. ($< 50\% \leadsto$ no bonus)
  - Bonus points $\widehat{=}$ percentage sum of the best 10 prepquizzes divided by 100.

Assessment, Grades

- **Overall (Module) Grade:**
  - Grade via the exam (Klausur) $\leadsto$ 100% of the grade.
  - Up to 10% bonus on-top for an exam with $\geq 50\%$ points. ($< 50\% \leadsto$ no bonus)
  - Bonus points $\widehat{=}$ percentage sum of the best 10 prepquizzes divided by 100.
- **Exam:** 60 minutes exam conducted in presence on paper ($\sim$ April 1. 2025)
- **Retake Exam:** 60 min exam six months later ($\sim$ October 1. 2025)
- ⚠ You have to register for exams in `https://campo.fau.de` in the first month of classes.
- **Note:** You can de-register from an exam on campo up to three working days before.

Preparedness Quizzes

- **PrepQuizzes:** Before every lecture we offer a 10 min online quiz – the PrepQuiz – about the material from the previous week. (10:00-10:10; starts in week 3)
- **Motivations:** We do this to
  - keep you prepared and working continuously. (primary)
  - bonus points if the exam has $\geq 50\%$ points (potential part of your grade)
  - update the ALeA learner model. (fringe benefit)
- The prepquiz will be given in the ALeA system



Question 1 of 2

04:42

(Minimum Function)
Let S an ordered set. Which statements about the minimum of S are true?

○ There is always a minimum in an ordered set.

● Each element x of the set S with $x \leq m$ for all $m \in S$ is a minimum of the set

○ Each element x of the set S with $x < m$ for all $m \in S$ is a minimum of the set

- `https://courses.voll-ki.fau.de/quiz-dash/lbs`
- You have to be logged into ALeA! (via FAU IDM)
- You can take the prepquiz on your laptop or phone, . . .
- . . . in the lecture or at home . . .
- . . . via WLAN or 4G Network. (do not overload)
- Prepquizzes will only be available 10:00-16:10!

Next Week: Pretest

- Next week we will try out the prepquiz infrastructure with a pretest!
  - **Presence**: bring your laptop or cellphone.
  - **Online**: you can and should take the pretest as well.
  - Have a recent firefox or chrome (chrome: younger than March 2023)
  - Make sure that you are logged into ALeA (via FAU IDM; see below)
- **Definition 1.1.** A pretest is an assessment for evaluating the preparedness of learners for further studies.
- **Concretely:** This pretest
  - establishes a baseline for the competency expectations in AI-1 and
  - tests the ALeA quiz infrastructure for the prepquizzes.
- Participation in the pretest is optional; it will not influence grades in any way.
- The pretest covers the prerequisites of AI-1 and some of the material that may have been covered in other courses.
- The test will be also used to refine the ALeA learner model, which may make learning experience in ALeA better. (see below)

1.2 Getting Most out of LBS

LBS Homework Assignments

- **Goal:** Homework assignments reinforce what was taught in lectures.
- **Homework Assignments:** Small individual problem/programming/proof task
  - but take time to solve (at least read them directly ⇝ questions)
- **Didactic Intuition:** Homework assignments give you material to test your understanding and show you how to apply it.
- ⚠ Homeworks give no points, but without trying you are unlikely to pass the exam.

LBS Homework Assignments

- **Goal:** Homework assignments reinforce what was taught in lectures.
- **Homework Assignments:** Small individual problem/programming/proof task
  - but take time to solve (at least read them directly ⇝ questions)
- **Didactic Intuition:** Homework assignments give you material to test your understanding and show you how to apply it.
- ⚠ Homeworks give no points, but without trying you are unlikely to pass the exam.
- Homeworks will be mainly peer-graded in the ALeA system.
- **Didactic Motivation:** Through peer grading students are able to see mistakes in their thinking and can correct any problems in future assignments. By grading assignments, students may learn how to complete assignments more accurately and how to improve their future results. (not just us being lazy)

LBS Homework Assignments – Howto

- **Homework Workflow:** in ALEA (see below)
  - Homework assignments will be published on thursdays: see `https://courses.voll-ki.fau.de/hw/lbs`
  - Submission of solutions via the ALEA system in the week after
  - Peer grading/feedback (and master solutions) via answer classes.
- **Quality Control:** TAs and instructors will monitor and supervise peer grading.

LBS Homework Assignments – Howto

- **Homework Workflow:** in ALeA (see below)
  - Homework assignments will be published on thursdays: see `https://courses.voll-ki.fau.de/hw/lbs`
  - Submission of solutions via the ALeA system in the week after
  - Peer grading/feedback (and master solutions) via answer classes.
- **Quality Control:** TAs and instructors will monitor and supervise peer grading.
- **Experiment:** Can we motivate enough of you to make peer assessment self-sustaining?
  - I am appealing to your sense of community responsibility here . . .
  - You should only expect other's to grade your submission if you grade their's (cf. Kant's "Moral Imperative")
  - **Make no mistake**: The grader usually learns at least as much as the gradee.

LBS Homework Assignments – Howto

- **Homework Workflow:** in ALeA (see below)
  - Homework assignments will be published on thursdays: see `https://courses.voll-ki.fau.de/hw/lbs`
  - Submission of solutions via the ALeA system in the week after
  - Peer grading/feedback (and master solutions) via answer classes.
- **Quality Control:** TAs and instructors will monitor and supervise peer grading.
- **Experiment:** Can we motivate enough of you to make peer assessment self-sustaining?
  - I am appealing to your sense of community responsibility here . . .
  - You should only expect other's to grade your submission if you grade their's (cf. Kant's "Moral Imperative")
  - **Make no mistake**: The grader usually learns at least as much as the gradee.
- **Homework/Tutorial Discipline:**
  - Start early! (many assignments need more than one evening's work)
  - Don't start by sitting at a blank screen (talking & study groups help)
  - Humans will be trying to understand the text/code/math when grading it.
  - Go to the tutorials, discuss with your TA! (they are there for you!)

Collaboration

- **Definition 2.1.** Collaboration (or cooperation) is the process of groups of agents acting together for common, mutual benefit, as opposed to acting in competition for selfish benefit. In a collaboration, every agent contributes to the common goal and benefits from the contributions of others.
- In learning situations, the benefit is “better learning”.
- **Observation:** In collaborative learning, the overall result can be significantly better than in competitive learning.
- **Good Practice:** Form study groups. (long- or short-term)
  1. ⚠ those learners who work most, learn most!
  2. ⚠ freeloaders – individuals who only watch – learn very little!
- It is OK to collaborate on homework assignments in LBS! (no bonus points)
- Choose your study group well! (We will (eventually) help via ALeA)

Do I need to attend the LBS Lectures

- Attendance is not mandatory for the LBS course. (official version)
- **Note:** There are two ways of learning: (both are OK, your mileage may vary)
  - Approach B: Read a book/papers (here: lecture notes)
  - Approach I: come to the lectures, be involved, interrupt the instructor whenever you have a question.

  The only advantage of I over B is that books/papers do not answer questions
- Approach S: come to the lectures and sleep does not work!
- The closer you get to research, the more we need to discuss!

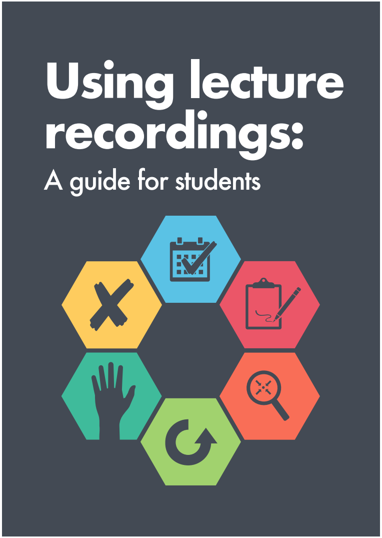
1.3 Learning Resources for AI-1

Textbook, Handouts and Information, Forums, Videos

- **(No) Textbook:** Lecture notes at `http://kwarc.info/teaching/LBS`
  - I mostly prepare them as we go along (semantically preloaded ⇝ research resource)
  - Please e-mail me any errors/shortcomings you notice. (improve for group)
- **For GLIF:** Frederik's Master's Thesis [Sch20]
- **Classical Semantics/Pragmatics:** (in the FAU Library)
  - Primary reference for LBS: [CKG09] (in the FAU Library)
  - also: [HHS07; Bir13; Rie10; ZS13; Sta14; Sae03; Por04; Kea11; Jac83; Cru11; Ari10]
- **Computational Semantics:** [BB05; EU10]
- **StudOn Forum:** `https://www.studon.fau.de/crs4625835.html` for
  - announcements, homeworks (my view on the forum)
  - questions, discussion among your fellow students (your forum too, use it!)
- **Course Videos:** at `https://www.fau.tv/course/id/4076.html`

Practical recommendations on Lecture Videos

- **Excellent Guide:** [Nor+18a] (German version at [Nor+18b])



Using lecture recordings:

A guide for students

- Attend lectures.
- Take notes.
- Be specific.
- Catch up.
- Ask for help.
- Don’t cut corners.

ALEA in LBS

- We assume that you already know the ALEA system from AI-1/2

Logic-Base Natural Language Semantics

This course covers the foundations of logic-based natural language processing — syntax, semantics construction, and pragmatics (semantic/pragmatic analysis) of natural language. On the one hand we will deveop the theoretical foundations (Montague's "method of fragments") and on the other hand we will implement severeal fragments in the GLIF system: a combination of the

- Grammatical Framework (GF),
- the MMT and MMT system for representing and processing formal systems, and
- the ELPI system, a higher-order logic programming language for mechanizing formal systems

and experiment with them.

As we only expect small class sizes, we will keep the course very interactive and project-oriented. The course 'Logic-basierte Representation for mathematical/technical Knowledge" (KRMT) from the previous summer semester is very helpful (as is LBS for KMRT) but not necessary.

The course is given in English.

ALeA in LBS

- We assume that you already know the ALeA system from AI-1/2
- Use it for
  - lecture notes (notes- vs slides-oriented)
  - flashcards (drill yourself on the LBS jargon/concepts)
  - course forum (questions, discussions and error reporting)
  - solving and peer-grading homework assignments
  - finding study groups (you need not endure LBS alone)
  - practicing with targeted problems (e.g. from old exams)
  - doing the prepquizzes (before each lecture)

Chapter 2
An Introduction to Natural Language Semantics

Fascination of (Natural) Language

- **Definition 0.1.** A natural language is any form of spoken or signed means of communication that has evolved naturally in humans through use and repetition without conscious planning or premeditation.
- **In other words:** the language you use all day long, e.g. English, German, ...
- **Why Should we care about natural language?:**
  - Even more so than thinking, language is a skill that only humans have.
  - It is a miracle that we can express complex thoughts in a sentence in a matter of seconds.
  - It is no less miraculous that a child can learn tens of thousands of words and complex syntax in a matter of a few years.

2.1 Natural Language and its Meaning

What is Natural Language Semantics? A Difficult Question!

- **Question:** What is “Natural Language Semantics”?

What is Natural Language Semantics? A Difficult Question!

- **Question:** What is "Natural Language Semantics"?
- **Definition 1.6 (Generic Answer).** Semantics is the study of reference, meaning, or truth.

What is Natural Language Semantics? A Difficult Question!

- **Question:** What is “Natural Language Semantics”?
- **Definition 1.11 (Generic Answer).** Semantics is the study of reference, meaning, or truth.
- **Definition 1.12.** A sign is anything that communicates a meaning that is not the sign itself to the interpreter of the sign. The meaning can be intentional, as when a word is uttered with a specific meaning, or unintentional, as when a symptom is taken as a sign of a particular medical condition
  Meaning is a relationship between signs and the objects they intend, express, or signify.
- **Definition 1.13.** Reference is a relationship between objects in which one object (the name) designates, or acts as a means by which to refer to – i.e. to connect to or link to – another object (the referent).
- **Definition 1.14.** Truth is the property of being in accord with reality in a/the mind-independent world. An object ascribed truth is called true, iff it is, and false, if it is not.

What is Natural Language Semantics? A Difficult Question!

- **Question:** What is "Natural Language Semantics"?
- **Definition 1.16 (Generic Answer).** Semantics is the study of reference, meaning, or truth.
- **Definition 1.17.** A sign is anything that communicates a meaning that is not the sign itself to the interpreter of the sign. The meaning can be intentional, as when a word is uttered with a specific meaning, or unintentional, as when a symptom is taken as a sign of a particular medical condition
  Meaning is a relationship between signs and the objects they intend, express, or signify.
- **Definition 1.18.** Reference is a relationship between objects in which one object (the name) designates, or acts as a means by which to refer to – i.e. to connect to or link to – another object (the referent).
- **Definition 1.19.** Truth is the property of being in accord with reality in a/the mind-independent world. An object ascribed truth is called true, iff it is, and false, if it is not.
- **Definition 1.20.** For natural language semantics, the signs are usually utterances and names are usually phrases.
- That is all very abstract and general, can we make this more concrete?
- Different (academic) disciplines find different concretizations.

What is (NL) Semantics? Answers from various Disciplines!

- **Observation:** Different (academic) disciplines specialize the notion of semantics (of natural language) in different ways.

What is (NL) Semantics? Answers from various Disciplines!

- **Observation:** Different (academic) disciplines specialize the notion of semantics (of natural language) in different ways.
- **Philosophy:** has a long history of trying to answer it, e.g.
  - Platon ⇝ cave allegory, Aristotle ⇝ Syllogisms.
  - Frege/Russell ⇝ sense vs. referent. (*Michael Kohlhase* vs. *Odysseus*)

What is (NL) Semantics? Answers from various Disciplines!

- **Observation:** Different (academic) disciplines specialize the notion of semantics (of natural language) in different ways.
- **Philosophy:** has a long history of trying to answer it, e.g.
  - Platon ⇝ cave allegory, Aristotle ⇝ Syllogisms.
  - Frege/Russell ⇝ sense vs. referent. (*Michael Kohlhase* vs. *Odysseus*)
- **Linguistics/Language Philosophy:** We need semantics e.g. in translation *Der Geist ist willig aber das Fleisch ist schwach!* vs. *Der Schnaps ist gut, aber der Braten ist verkocht!* (meaning counts)

What is (NL) Semantics? Answers from various Disciplines!

- **Observation:** Different (academic) disciplines specialize the notion of semantics (of natural language) in different ways.
- **Philosophy:** has a long history of trying to answer it, e.g.
  - Platon $\leadsto$ cave allegory, Aristotle $\leadsto$ Syllogisms.
  - Frege/Russell $\leadsto$ sense vs. referent. (*Michael Kohlhase vs. Odysseus*)
- **Linguistics/Language Philosophy:** We need semantics e.g. in translation *Der Geist ist willig aber das Fleisch ist schwach!* vs. *Der Schnaps ist gut, aber der Braten ist verkocht!* (meaning counts)
- **Psychology/Cognition:** Semantics $\widehat{=}$ "what is in our brains" ($\leadsto$ mental models)

What is (NL) Semantics? Answers from various Disciplines!

- **Observation:** Different (academic) disciplines specialize the notion of semantics (of natural language) in different ways.
- **Philosophy:** has a long history of trying to answer it, e.g.
  - Platon ⇝ cave allegory, Aristotle ⇝ Syllogisms.
  - Frege/Russell ⇝ sense vs. referent. (*Michael Kohlhase vs. Odysseus*)
- **Linguistics/Language Philosophy:** We need semantics e.g. in translation *Der Geist ist willig aber das Fleisch ist schwach!* vs. *Der Schnaps ist gut, aber der Braten ist verkocht!* (meaning counts)
- **Psychology/Cognition:** Semantics $\widehat{=}$ "what is in our brains" (⇝ mental models)
- **Mathematics** has driven much of modern logic in the quest for foundations.
  - Logic as "foundation of mathematics" solved as far as possible
  - In daily practice syntax and semantics are not differentiated (much).

What is (NL) Semantics? Answers from various Disciplines!

- **Observation:** Different (academic) disciplines specialize the notion of semantics (of natural language) in different ways.
- **Philosophy:** has a long history of trying to answer it, e.g.
  - Platon $\leadsto$ cave allegory, Aristotle $\leadsto$ Syllogisms.
  - Frege/Russell $\leadsto$ sense vs. referent. (*Michael Kohlhase vs. Odysseus*)
- **Linguistics/Language Philosophy:** We need semantics e.g. in translation *Der Geist ist willig aber das Fleisch ist schwach!* vs. *Der Schnaps ist gut, aber der Braten ist verkocht!* (meaning counts)
- **Psychology/Cognition:** Semantics $\widehat{=}$ "what is in our brains" ($\leadsto$ mental models)
- **Mathematics** has driven much of modern logic in the quest for foundations.
  - Logic as "foundation of mathematics" solved as far as possible
  - In daily practice syntax and semantics are not differentiated (much).
- **Logic@AI/CS** tries to define meaning and compute with them. (applied semantics)
  - makes syntax explicit in a formal language (formulae, sentences)
  - defines truth/validity by mapping sentences into "world" (interpretation)
  - gives rules of truth-preserving reasoning (inference)

Meaning of Natural Language; e.g. Machine Translation

- **Idea:** Machine translation is very simple! (we have good lexica)
- **Example 1.21.** *Peter liebt Maria.* $\rightsquigarrow$ *Peter loves Mary.*
- ⚠ this only works for simple examples!
- **Example 1.22.** *Wirf der Kuh das Heu über den Zaun.* $\not\rightsquigarrow$ *Throw the cow the hay over the fence.* (differing grammar; Google Translate)
- **Example 1.23.** ⚠ Grammar is not the only problem
  - *Der Geist ist willig, aber das Fleisch ist schwach!*
  - *Der Schnaps ist gut, aber der Braten ist verkocht!*
- **Observation 1.24.** *We have to understand the meaning for high-quality translation!*

Language and Information

- **Observation:** Humans use words (sentences, texts) in natural languages to represent and communicate information.
- **But:** What really counts is not the words themselves, but the meaning information they carry.

Language and Information

- **Observation:** Humans use words (sentences, texts) in natural languages to represent and communicate information.
- **But:** What really counts is not the words themselves, but the meaning information they carry.
- **Example 1.27 (Word Meaning).**

Newspaper ⇝

The New York Times

NEW YORK, THURSDAY, NOVEMBER 11, 2004

RECITAL REVIEW

Offering Beethoven With Crisp Lines, but Softened by 'Moonlight'

By ALLAN KOZINN

Soheil Nasseri

Alice Tully Hall

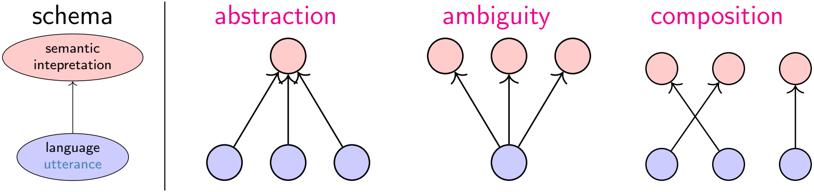
- For questions/answers, it would be very useful to find out what words (sentences/texts) mean.

Language and Information

- **Observation:** Humans use words (sentences, texts) in natural languages to represent and communicate information.
- **But:** What really counts is not the words themselves, but the meaning information they carry.
- **Example 1.29 (Word Meaning).**

Newspaper ⇝

- For questions/answers, it would be very useful to find out what words (sentences/texts) mean.
- **Definition 1.30.** Interpretation of natural language utterances: three problems



schema

semantic intepretation

language utterance

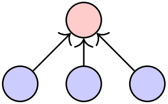
abstraction

ambiguity

composition

Language and Information (Examples)

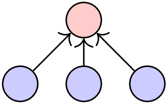
- **Example 1.31 (Abstraction).**



Car and *automobile* have the same meaning.

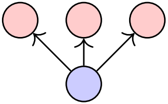
Language and Information (Examples)

- **Example 1.34 (Abstraction).**



Car and *automobile* have the same meaning.

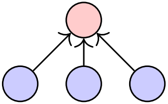
- **Example 1.35 (Ambiguity).**



A *bank* can be a financial institution or a geographical feature.

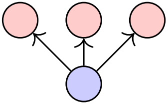
Language and Information (Examples)

- **Example 1.37 (Abstraction).**



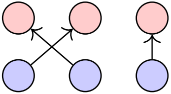
Car and *automobile* have the same meaning.

- **Example 1.38 (Ambiguity).**



A *bank* can be a financial institution or a geographical feature.

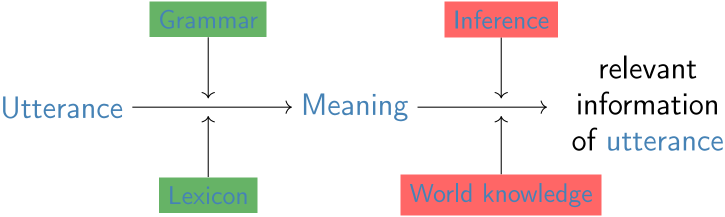
- **Example 1.39 (Composition).**



Every student sleeps $\leadsto \forall x.\mathit{student}(x) \Rightarrow \mathit{sleep}(x)$

Context Contributes to the Meaning of NL Utterances

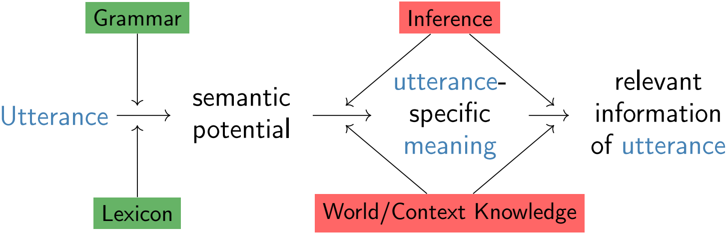
- **Observation:** Not all information conveyed is linguistically realized in an utterance.
- **Example 1.40.** *The lecture begins at 11:00 am.* What lecture? Today?
- **Definition 1.41.** We call a piece i of information linguistically realized in an utterance U, iff, we can trace i to a fragment of U.
- **Definition 1.42 (Possible Mechanism).** Inferring the missing pieces from the context and world knowledge:



We call this process semantic/pragmatic analysis.

Context Contributes to the Meaning of NL Utterances

- **Example 1.43.** *It starts at eleven.* What starts?
- Before we can resolve the time, we need to resolve the anaphor *it*.
- **Possible Mechanism:** More Inference!



Grammar

Lexicon

Utterance

semantic potential

Inference

World/Context Knowledge

utterance-specific meaning

relevant information of utterance

⇝ Semantic/pragmatic analysis is quite complex! (prime topic of LBS)

Semantics is not a Cure-It-All!

How many animals of each species did Moses take onto the ark?

Semantics is not a Cure-It-All!

How many animals of each species did Moses take onto the ark?

- Actually, it was Noah (But you understood the question anyways)

But Semantics works in some cases

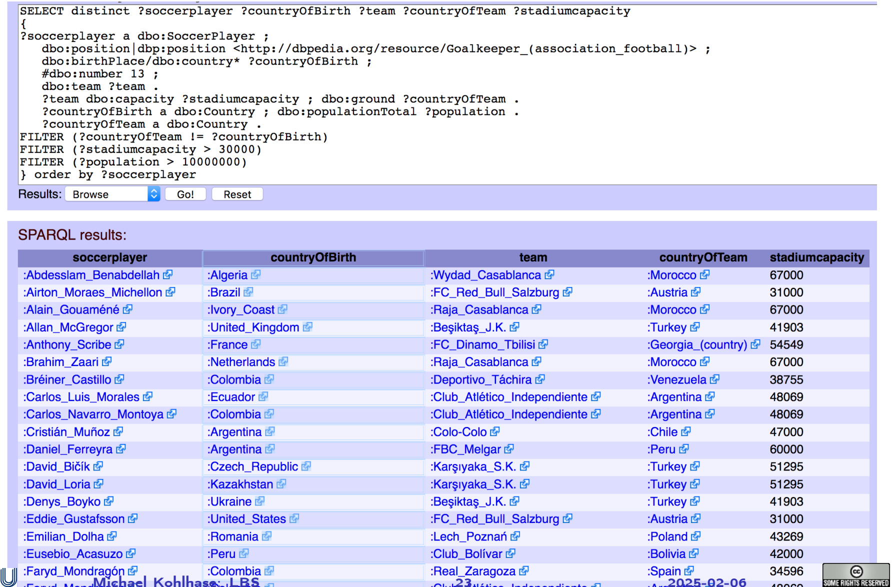
- The only thing that currently really helps is a restricted domain:
  - I. e. a restricted vocabulary and world model.

But Semantics works in some cases

- The only thing that currently really helps is a restricted domain:
  - I. e. a restricted vocabulary and world model.
- **Demo:**
  DBPedia `http://dbpedia.org/snorql/`
  Query: Soccer players, who are born in a country with more than 10 million inhabitants, who played as goalkeeper for a club that has a stadium with more than 30.000 seats and the club country is different from the birth country

But Semantics works in some cases

- **Answer:**
  (is computed by DBPedia from a SPARQL query)



```
SELECT distinct ?soccerplayer ?countryOfBirth ?team ?countryOfTeam ?stadiumcapacity
{
?soccerplayer a dbo:SoccerPlayer ;
   dbo:position|dbp:position <http://dbpedia.org/resource/Goalkeeper_(association_football)> ;
   dbo:birthPlace/dbo:country* ?countryOfBirth ;
   #dbo:number 13 ;
   dbo:team ?team .
   ?team dbo:capacity ?stadiumcapacity ; dbo:ground ?countryOfTeam .
   ?countryOfBirth a dbo:Country ; dbo:populationTotal ?population .
   ?countryOfTeam a dbo:Country .
FILTER (?countryOfTeam != ?countryOfBirth)
FILTER (?stadiumcapacity > 30000)
FILTER (?population > 10000000)
} order by ?soccerplayer
```

SPARQL results:

soccerplayer	countryOfBirth	team	countryOfTeam	stadiumcapacity
:Abdesslam_Benabdellah	:Algeria	:Wydad_Casablanca	:Morocco	67000
:Airton_Moraes_Michellon	:Brazil	:FC_Red_Bull_Salzburg	:Austria	31000
:Alain_Gouaméné	:Ivory_Coast	:Raja_Casablanca	:Morocco	67000
:Allan_McGregor	:United_Kingdom	:Beşiktaş_J.K.	:Turkey	41903
:Anthony_Scribe	:France	:FC_Dinamo_Tbilisi	:Georgia_(country)	54549
:Brahim_Zaari	:Netherlands	:Raja_Casablanca	:Morocco	67000
:Bréiner_Castillo	:Colombia	:Deportivo_Táchira	:Venezuela	38755
:Carlos_Luis_Morales	:Ecuador	:Club_Atlético_Independiente	:Argentina	48069
:Carlos_Navarro_Montoya	:Colombia	:Club_Atlético_Independiente	:Argentina	48069
:Cristián_Muñoz	:Argentina	:Colo-Colo	:Chile	47000
:Daniel_Ferreyra	:Argentina	:FBC_Melgar	:Peru	60000
:David_Bičík	:Czech_Republic	:Karşıyaka_S.K.	:Turkey	51295
:David_Loria	:Kazakhstan	:Karşıyaka_S.K.	:Turkey	51295
:Denys_Boyko	:Ukraine	:Beşiktaş_J.K.	:Turkey	41903
:Eddie_Gustafsson	:United_States	:FC_Red_Bull_Salzburg	:Austria	31000
:Emilian_Dolha	:Romania	:Lech_Poznań	:Poland	43269
:Eusebio_Acasuzo	:Peru	:Club_Bolívar	:Bolivia	42000
:Faryd_Mondragón	:Colombia	:Real_Zaragoza	:Spain	34596

2.2 Natural Language Understanding as Engineering

Language Technology

- Language Assistance:
  - written language: Spell/grammar/style-checking,
  - spoken language: dictation systems and screen readers,
  - multilingual text: machine-supported text and dialog translation, eLearning.

Language Technology

- Language Assistance:
  - written language: Spell/grammar/style-checking,
  - spoken language: dictation systems and screen readers,
  - multilingual text: machine-supported text and dialog translation, eLearning.
- Information management:
  - search and classification of documents, (e.g. Google/Bing)
  - information extraction, question answering. (e.g. `http://ask.com`)

Language Technology

- Language Assistance:
  - written language: Spell/grammar/style-checking,
  - spoken language: dictation systems and screen readers,
  - multilingual text: machine-supported text and dialog translation, eLearning.
- Information management:
  - search and classification of documents, (e.g. Google/Bing)
  - information extraction, question answering. (e.g. `http://ask.com`)
- Dialog Systems/Interfaces:
  - information systems: at airport, tele-banking, e-commerce, call centers,
  - dialog interfaces for computers, robots, cars. (e.g. Siri/Alexa)

Language Technology

- Language Assistance:
  - written language: Spell/grammar/style-checking,
  - spoken language: dictation systems and screen readers,
  - multilingual text: machine-supported text and dialog translation, eLearning.
- Information management:
  - search and classification of documents, (e.g. Google/Bing)
  - information extraction, question answering. (e.g. `http://ask.com`)
- Dialog Systems/Interfaces:
  - information systems: at airport, tele-banking, e-commerce, call centers,
  - dialog interfaces for computers, robots, cars. (e.g. Siri/Alexa)
- **Observation:** The earlier technologies largely rely on pattern matching, the latter ones need to compute the meaning of the input utterances, e.g. for database lookups in information systems.

What is Natural Language Processing?

- **Generally:** Studying of natural languages and development of systems that can use/generate these.
- **Definition 2.1.** Natural language processing (NLP) is an engineering field at the intersection of computer science, artificial intelligence, and linguistics which is concerned with the interactions between computers and human (natural) languages. Most challenges in NLP involve:
  - Natural language understanding (NLU) that is, enabling computers to derive meaning (representations) from human or natural language input.
  - Natural language generation (NLG) which aims at generating natural language or speech from meaning representation.
- For communication with/among humans we need both NLU and NLG.

What is the State of the Art In NLU?

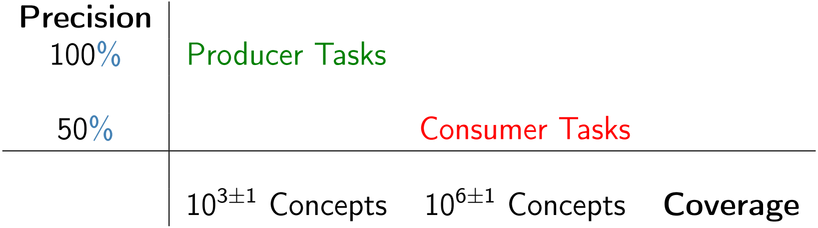
- Two avenues of attack for the problem: knowledge-based and statistical techniques (they are complementary)

Deep	Knowledge-based We are here	Not there yet cooperation?
Shallow	no-one wants this	Statistical Methods applications
Analysis ↑ vs. Coverage →	narrow	wide

- We will cover foundational methods of deep processing in the course and a mixture of deep and shallow ones in the lab.

Environmental Niches for both Approaches to NLU

- **Definition 2.2.** There are two kinds of applications/tasks in NLU:
  - Consumer tasks: consumer grade applications have tasks that must be fully generic and wide coverage. (e.g. machine translation like Google Translate)
  - Producer tasks: producer grade applications must be high-precision, but can be domain-specific (e.g. multilingual documentation, machinery-control, program verification, medical technology)



Precision

100% Producer Tasks

50% Consumer Tasks

$10^{3\pm1}$ Concepts $10^{6\pm1}$ Concepts **Coverage**

after Aarne Ranta [Ran17].

- **Example 2.3.** Producing/managing machine manuals in multiple languages across machine variants is a critical producer task for machine tool company.
- A producer domain I am interested in: mathematical/technical documents.

NLP for NLU: The Waterfall Model

- **Definition 2.4 (The NLU Waterfall).** NL understanding is often modeled as a simple linear process: the NLU waterfall consists of five consecutive steps:
  0) speech processing: acoustic signal $\rightsquigarrow$ word hypothesis graph
  1) syntactic processing: word sequence $\rightsquigarrow$ phrase structure
  2) semantics construction: phrase structure $\rightsquigarrow$ (quasi-)logical form
  3) semantic/pragmatic analysis:
     (quasi-)logical form $\rightsquigarrow$ knowledge representation
  4) problem solving: using the generated knowledge (application-specific)
- **Definition 2.5.** We call any formalization of an utterance as a logical formula a logical form. A quasi-logical form (QLF) is a representation which can be turned into a logical form by further computation.2
- **In this course:** steps 1), 2) and 3).

2.3 Looking at Natural Language

Fun with Diamonds (are they real?) [Dav67b]

- **Example 3.1.** We study the truth conditions of adjectival complexes:
  - *This is a diamond.* ($\models$ *diamond*)

Fun with Diamonds (are they real?) [Dav67b]

- **Example 3.2.** We study the truth conditions of adjectival complexes:
  - *This is a diamond.* ($\models$ *diamond*)
  - *This is a blue diamond.* ($\models$ *diamond*, $\models$ *blue*)

Fun with Diamonds (are they real?) [Dav67b]

- **Example 3.3.** We study the truth conditions of adjectival complexes:
  - *This is a diamond.* ($\models$ *diamond*)
  - *This is a blue diamond.* ($\models$ *diamond*, $\models$ *blue*)
  - *This is a big diamond.* ($\models$ *diamond*, $\not\models$ *big*)

Fun with Diamonds (are they real?) [Dav67b]

- **Example 3.4.** We study the truth conditions of adjectival complexes:
  - *This is a diamond.* ($\models$ *diamond*)
  - *This is a blue diamond.* ($\models$ *diamond*, $\models$ *blue*)
  - *This is a big diamond.* ($\models$ *diamond*, $\not\models$ *big*)
  - *This is a fake diamond.* ($\models \neg$*diamond*)

Fun with Diamonds (are they real?) [Dav67b]

- **Example 3.5.** We study the truth conditions of adjectival complexes:
  - *This is a diamond.* $(\models diamond)$
  - *This is a blue diamond.* $(\models diamond, \models blue)$
  - *This is a big diamond.* $(\models diamond, \not\models big)$
  - *This is a fake diamond.* $(\models \neg diamond)$
  - *This is a fake blue diamond.* $(\models blue?, \models diamond?)$

Fun with Diamonds (are they real?) [Dav67b]

- **Example 3.6.** We study the truth conditions of adjectival complexes:
  - *This is a diamond.* ($\models$ *diamond*)
  - *This is a blue diamond.* ($\models$ *diamond*, $\models$ *blue*)
  - *This is a big diamond.* ($\models$ *diamond*, $\not\models$ *big*)
  - *This is a fake diamond.* ($\models \neg$*diamond*)
  - *This is a fake blue diamond.* ($\models$ *blue*?, $\models$ *diamond*?)
  - *Mary knows that this is a diamond.* ($\models$ *diamond*)

Fun with Diamonds (are they real?) [Dav67b]

- **Example 3.7.** We study the truth conditions of adjectival complexes:
  - *This is a diamond.* ($\models$ *diamond*)
  - *This is a blue diamond.* ($\models$ *diamond*, $\models$ *blue*)
  - *This is a big diamond.* ($\models$ *diamond*, $\not\models$ *big*)
  - *This is a fake diamond.* ($\models \neg$*diamond*)
  - *This is a fake blue diamond.* ($\models$ *blue?*, $\models$ *diamond?*)
  - *Mary knows that this is a diamond.* ($\models$ *diamond*)
  - *Mary believes that this is a diamond.* ($\not\models$ *diamond*)

Ambiguity: The dark side of Meaning

- **Definition 3.8.** We call an utterance ambiguous, iff it has multiple meanings, which we call readings.
- **Example 3.9.** All of the following sentences are ambiguous:
  - *John went to the bank.* (river or financial?)

Ambiguity: The dark side of Meaning

- **Definition 3.10.** We call an utterance ambiguous, iff it has multiple meanings, which we call readings.
- **Example 3.11.** All of the following sentences are ambiguous:
  - *John went to the bank.* (river or financial?)
  - *You should have seen the bull we got from the pope.* (three readings!)

Ambiguity: The dark side of Meaning

- **Definition 3.12.** We call an utterance ambiguous, iff it has multiple meanings, which we call readings.
- **Example 3.13.** All of the following sentences are ambiguous:
  - *John went to the bank.* (river or financial?)
  - *You should have seen the bull we got from the pope.* (three readings!)
  - *I saw her duck.* (animal or action?)

Ambiguity: The dark side of Meaning

- **Definition 3.14.** We call an utterance ambiguous, iff it has multiple meanings, which we call readings.
- **Example 3.15.** All of the following sentences are ambiguous:
  - *John went to the bank.* (river or financial?)
  - *You should have seen the bull we got from the pope.* (three readings!)
  - *I saw her duck.* (animal or action?)
  - *John chased the gangster in the red sports car.* (three-way too!)

Quantifiers, Scope and Context

- **Example 3.16.** *Every man loves a woman.* (Keira Knightley or his mother!)

Quantifiers, Scope and Context

- **Example 3.21.** *Every man loves a woman.* (Keira Knightley or his mother!)
- **Example 3.22.** *Every car has a radio.* (only one reading!)

Quantifiers, Scope and Context

- **Example 3.26.** *Every man loves a woman.* (Keira Knightley or his mother!)
- **Example 3.27.** *Every car has a radio.* (only one reading!)
- **Example 3.28.** *Some student in every course sleeps in every class at least some of the time.* (how many readings?)

Quantifiers, Scope and Context

- **Example 3.31.** *Every man loves a woman.* (Keira Knightley or his mother!)
- **Example 3.32.** *Every car has a radio.* (only one reading!)
- **Example 3.33.** *Some student in every course sleeps in every class at least some of the time.* (how many readings?)
- **Example 3.34.** *The president of the US is having an affair with an intern.* (2002 or 2000?)

Quantifiers, Scope and Context

- **Example 3.36.** *Every man loves a woman.* (Keira Knightley or his mother!)
- **Example 3.37.** *Every car has a radio.* (only one reading!)
- **Example 3.38.** *Some student in every course sleeps in every class at least some of the time.* (how many readings?)
- **Example 3.39.** *The president of the US is having an affair with an intern.* (2002 or 2000?)
- **Example 3.40.** *Everyone is here.* (who is everyone?)

More Context: Anaphora – Challenge for Pragmatic Analysis

- **Example 3.41 (Anaphoric References).**
  - *John is a bachelor. His wife is very nice.* (Uh, what?, who?)

More Context: Anaphora – Challenge for Pragmatic Analysis

- **Example 3.45 (Anaphoric References).**
  - *John is a bachelor. His wife is very nice.* (Uh, what?, who?)
  - *John likes his dog Spiff even though he bites him sometimes.* (who bites?)

More Context: Anaphora – Challenge for Pragmatic Analysis

- **Example 3.49 (Anaphoric References).**
  - *John is a bachelor. His wife is very nice.* (Uh, what?, who?)
  - *John likes his dog Spiff even though he bites him sometimes.* (who bites?)
  - *John likes Spiff. Peter does too.* (what to does Peter do?)

More Context: Anaphora – Challenge for Pragmatic Analysis

- **Example 3.53 (Anaphoric References).**
  - *John is a bachelor. His wife is very nice.* (Uh, what?, who?)
  - *John likes his dog Spiff even though he bites him sometimes.* (who bites?)
  - *John likes Spiff. Peter does too.* (what to does Peter do?)
  - *John loves his wife. Peter does too.* (whom does Peter love?)

More Context: Anaphora – Challenge for Pragmatic Analysis

- **Example 3.57 (Anaphoric References).**
  - *John is a bachelor. His wife is very nice.* (Uh, what?, who?)
  - *John likes his dog Spiff even though he bites him sometimes.* (who bites?)
  - *John likes Spiff. Peter does too.* (what to does Peter do?)
  - *John loves his wife. Peter does too.* (whom does Peter love?)
  - *John loves golf, and Mary too.* (who does what?)

More Context: Anaphora – Challenge for Pragmatic Analysis

- **Example 3.61 (Anaphoric References).**
  - *John is a bachelor. His wife is very nice.* (Uh, what?, who?)
  - *John likes his dog Spiff even though he bites him sometimes.* (who bites?)
  - *John likes Spiff. Peter does too.* (what to does Peter do?)
  - *John loves his wife. Peter does too.* (whom does Peter love?)
  - *John loves golf, and Mary too.* (who does what?)
- **Definition 3.62.** A word or phrase is called anaphoric (or an anaphor), if its interpretation depends upon another phrase in context. In a narrower sense, an anaphor refers to an earlier phrase (its antecedent), while a cataphor to a later one (its postcedent).

  Definition 3.63. The process of determining the antecedent or postcedent of an anaphoric phrase is called anaphor resolution.

  Definition 3.64. An anaphoric connection between anaphor and its antecedent or postcedent is called direct, iff it can be understood purely syntactically. An anaphoric connection is called indirect or a bridging reference if additional knowledge is needed.

More Context: Anaphora – Challenge for Pragmatic Analysis

- **Example 3.65 (Anaphoric References).**
  - *John is a bachelor. His wife is very nice.* (Uh, what?, who?)
  - *John likes his dog Spiff even though he bites him sometimes.* (who bites?)
  - *John likes Spiff. Peter does too.* (what to does Peter do?)
  - *John loves his wife. Peter does too.* (whom does Peter love?)
  - *John loves golf, and Mary too.* (who does what?)
- **Definition 3.66.** A word or phrase is called anaphoric (or an anaphor), if its interpretation depends upon another phrase in context. In a narrower sense, an anaphor refers to an earlier phrase (its antecedent), while a cataphor to a later one (its postcedent).

  Definition 3.67. The process of determining the antecedent or postcedent of an anaphoric phrase is called anaphor resolution.

  Definition 3.68. An anaphoric connection between anaphor and its antecedent or postcedent is called direct, iff it can be understood purely syntactically. An anaphoric connection is called indirect or a bridging reference if additional knowledge is needed.
- Anaphora are another example, where natural languages use the inferential capabilities of the hearer/reader to "shorten" utterances.
- Anaphora challenge pragmatic analysis, since they can only be resolved from the context using world knowledge.

Context is Personal and Keeps Changing

- **Example 3.69.** Consider the following sentences involving definite description:
  1. *The king of America is rich.* (true or false?)

How do the interact with your context and world knowledge?

Context is Personal and Keeps Changing

- **Example 3.70.** Consider the following sentences involving definite description:
  1. *The king of America is rich.* (true or false?)
  2. *The king of America isn't rich.* (false or true?)

How do the interact with your context and world knowledge?

Context is Personal and Keeps Changing

- **Example 3.71.** Consider the following sentences involving definite description:
  1. *The king of America is rich.* (true or false?)
  2. *The king of America isn't rich.* (false or true?)
  3. *If America had a king, the king of America would be rich.* (true or false!)

  How do the interact with your context and world knowledge?

Context is Personal and Keeps Changing

- **Example 3.72.** Consider the following sentences involving definite description:
  1. *The king of America is rich.* (true or false?)
  2. *The king of America isn't rich.* (false or true?)
  3. *If America had a king, the king of America would be rich.* (true or false!)
  4. *The king of Buganda is rich.* (Where is Buganda?)

  How do the interact with your context and world knowledge?

Context is Personal and Keeps Changing

- **Example 3.73.** Consider the following sentences involving definite description:
  1. *The king of America is rich.* (true or false?)
  2. *The king of America isn't rich.* (false or true?)
  3. *If America had a king, the king of America would be rich.* (true or false!)
  4. *The king of Buganda is rich.* (Where is Buganda?)
  5. *… Joe Smith… The CEO of Westinghouse announced budget cuts.* (CEO=J.S.!)

  How do the interact with your context and world knowledge?

Context is Personal and Keeps Changing

- **Example 3.74.** Consider the following sentences involving definite description:
  1. *The king of America is rich.* (true or false?)
  2. *The king of America isn't rich.* (false or true?)
  3. *If America had a king, the king of America would be rich.* (true or false!)
  4. *The king of Buganda is rich.* (Where is Buganda?)
  5. *. . . Joe Smith. . . The CEO of Westinghouse announced budget cuts.* (CEO=J.S.!)

  How do the interact with your context and world knowledge?
- The interpretation or whether they make sense at all dep
- **Note:** Last two examples feed back into the context or even world knowledge:
  - If 4. is uttered by an Africa expert, we add "*Buganda exists and is a monarchy* to our world knowledge
  - We add *Joe Smith is the CEO of Westinghouse to the context/world knowledge* (happens all the time in newpaper articles)

2.4 A Taste of Language Philosophy

What is the Meaning of Natural Language Utterances?

- **Question:** What is the meaning of the word *chair*?

What is the Meaning of Natural Language Utterances?

- **Question:** What is the meaning of the word *chair*?
- **Answer:** “the set of all chairs” (difficult to delineate, but more or less clear)
- **Question:** What is the meaning of the word *Michael Kohlhase*?

What is the Meaning of Natural Language Utterances?

- **Question:** What is the meaning of the word *chair*?
- **Answer:** “the set of all chairs” (difficult to delineate, but more or less clear)
- **Question:** What is the meaning of the word *Michael Kohlhase*?
- **Answer:** The word refers to an object in the real world: the instructor of LBS.
- **Alternatively:** The singleton with that object (as for “set of chairs” above)
- **Question:** What about *Michael Kohlhase sits on a chair*?

What is the Meaning of Natural Language Utterances?

- **Question:** What is the meaning of the word *chair*?
- **Answer:** "the set of all chairs" (difficult to delineate, but more or less clear)
- **Question:** What is the meaning of the word *Michael Kohlhase*?
- **Answer:** The word refers to an object in the real world: the instructor of LBS.
- **Alternatively:** The singleton with that object (as for "set of chairs" above)
- **Question:** What about *Michael Kohlhase sits on a chair*?
- **Towards an Answer:** We have to combine the two sets, via the meaning of "sits".
- **Question:** What is the meaning of the word *John F. Kennedy* or *Odysseus*?

What is the Meaning of Natural Language Utterances?

- **Question:** What is the meaning of the word *chair*?
- **Answer:** "the set of all chairs" (difficult to delineate, but more or less clear)
- **Question:** What is the meaning of the word *Michael Kohlhase*?
- **Answer:** The word refers to an object in the real world: the instructor of LBS.
- **Alternatively:** The singleton with that object (as for "set of chairs" above)
- **Question:** What about *Michael Kohlhase sits on a chair*?
- **Towards an Answer:** We have to combine the two sets, via the meaning of "sits".
- **Question:** What is the meaning of the word *John F. Kennedy* or *Odysseus*?
- **Problem:** There are no objects in the real worlds, so the meaning of both is $\emptyset$ and thus equal ☹.

2.4.1 Epistemology: The Philosphy of Science

Epistemology – Propositions & Observations

- **Definition 4.1.** Epistemology is the branch of philosophy concerned with studying nature of knowledge, its justification, the rationality of belief, scientific theories and predictions, and various related issues.
- **Definition 4.2.** A proposition is a sentence about the actual world or a class of worlds deemed possible whose meaning can be expressed as being true or false in a specific world.
- **Definition 4.3.** A belief is a proposition φ that an agent a holds true about a class of worlds. This is a characterizing feature of the agent.
- **Definition 4.4 (Knowledge - The JTB Account).** Knowledge is justified, true belief.
- **Problem:** How can an agent justify a belief to obtain knowledge.
- **Definition 4.5.** Given a world w, the observed value (or just value, i.e. true or false) of a proposition (in w) can be determined by observations, that is an agent, the observer, either observes (experiences) that φ is true in w or conducts a deliberate, systematic experiment that determines φ to be true in w.

Epistemology – Reproducibility & Phenomena

- **Problem:** Observations are sometimes unreliable, e.g. observer o perceives φ to be true, while it is false or vice versa.
- **Idea:** Repeat the observations to raise the probability of getting them right.
- **Definition 4.6.** An observation φ is said to be reproducible, iff φ can observed by different observers in different situations.
- **Definition 4.7.** A phenomenon φ is a proposition that is reproducibly observable to be true in a class of worlds.
- **Problem:** We would like to verify a phenomenon φ, i.e. observe φ in all worlds, But relevant world classes are too large to make this practically feasible.
- **Definition 4.8.** A world w is a counterexample to a proposition φ, if φ is observably false in w.
- **Intuition:** The absence of counterexamples is the best we can hope for in general for accepting phenomena.
- **Intuition:** The phenomena constitute the "world model" of an agent.
- **Problem:** It is impossible/inefficient (for an agent) to know all phenomena.
- **Idea:** An agent could retain only a small subset of known propositions, from this all phenomena can be derived.

Epistemology – Explanations & Hypotheses

- **Definition 4.9.** A proposition ψ follows from a proposition φ, iff ψ is true in any world where φ is.
- **Definition 4.10.** An explanation of a phenomenon φ is a set Φ of propositions, such that φ follows from Φ.
- **Example 4.11.** $\{\varphi\}$ is a (rather useless) explanation for φ.
- **Intuition:** We prefer explanations Φ that explain more than just φ.
- **Observation:** This often coincides with explanations that are in some sense "simpler" or "more elementary" than φ. ($\rightsquigarrow$ Occam's razor)
- **Definition 4.12.** A proposition is called falsifiable, iff counterexamples are theoretically possible and the observation of a reproducible series of counterexample is practically feasible.
- **Definition 4.13.** A hypothesis is a proposed explanation of a phenomenon that is falsifiable.

Epistemology – Scientific Theories

- **Knowledge Strategy:** Collect hypotheses about the world, drop those with counterexamples and those that can be explained themselves.
- **Definition 4.14.** A hypothesis φ can be tested in world/situation w by observing the value of φ in w. If the value is true, then we say that the observation o supports φ or is evidence for φ. If it is false then o falsifies φ.
- **Definition 4.15.** A (scientific) theory for a collection Φ of phenomena is a set Θ of hypotheses that
  - has been tested extensively and rigorously without finding counterexamples, and
  - is minimal in the sense that no sub-collection of Θ explains Φ.
- **Definition 4.16.** We call any proposition φ that follows from a theory Φ a prediction of Φ.
- **Note:** To falsify a theory Φ, it is sufficient to falsify any prediction. Any observation of a prediction φ of Φ supports Φ.

2.4.2 Meaning Theories

Theories of Meaning

- **The Central Question:** What is the meaning of natural language?
- This is difficult to answer definitely, . . .
- **But** we can form meaning theory that make predictions that we can test.
- **Definition 4.17.** A semantic meaning theory assigns semantic contents to expressions of a language.
- **Definition 4.18.** A foundational meaning theory tries to explain why language expressions have the meanings they have; e.g. in terms of mental states of individuals and groups.
- It is important to keep these two notions apart.
- We will concentrate on semantic meaning theories in this course.

The Meaning of Singular Terms

- Let's see a semantic meaning theory in action.
- **Definition 4.19.** A singular term is a phrase that purports to denote or designate a particular individual person, place, or other object.
- **Example 4.20.** *Michael Kohlhase* and *Odysseus* are singular terms.
- **Definition 4.21.** In [Fre92], Gottlob Frege distinguishes between sense (Sinn) and referent (Bedeutung) of singular terms.
- **Example 4.22.** Even though *Odysseus* does not have a referent, it has a very real sense. (but what is a sense?)
- **Example 4.23.** The ancient greeks knew the planets *Hesperos* (the evening star) and *Phosphoros* (the morning star). These words have different senses, but the – as we now know – same referent: the planet Venus.
- **Remark:** Bertrand Russell views singular terms as disguised definite descriptions – *Hesperos* as "the brightest heavenly body that sometimes rises in the evening". Frege's sense can often be conflated with Russell's descriptions. (there can be more than one definite description)

Cresswell's "Most Certain Principle" and Truth Conditions

- **Problem:** How can we test meaning theories in practice?
- **Definition 4.24.** Cresswell's (1982) most certain principle (MCP): [Cre82]

  > I'm going to begin by telling you what I think is the most certain thing I think about meaning. Perhaps it's the only thing. It is this. If we have two sentences A and B, and A is true and B is false, then A and B do not mean the same.

- **Definition 4.25.** The truth conditions of a sentence are the conditions of the world under which it is true. These conditions must be such that if all obtain, the sentence is true, and if one doesn't obtain, the sentence is false.
- **Observation:** Meaning determines truth conditions and vice versa.
- **In Fregean terms** The sense of a sentence (a thought) determines its referent (a truth value).

This principle sounds trivial – and indeed it is, if you think about it – but gives rise to the notion of truth conditions, which form the most important way of finding out about the meaning of sentences: the determinations of truth conditions.

Truth Conditions in Practice

- **Idea:** To test/determine the truth conditions of a sentence S in practice, we tell little stories that describe situations/worlds that embed S.
- **Example 4.26.** Consider the ambiguous sentence from Example 3.27 (Looking at Natural Language) in the LBS lecture notes:
  John chased the gangster in the red sports car.
  For each of three readings there is story $\widehat{=}$ truth conditions
  - John drives the red sports car and chases the gangster.
  - John chases the gangster who drives the red sports car.
  - John chases the gangster on the back seat of a (very very big) red sports car.

  All of these stories correspond to different worlds, so by the MCP there must be at least three readings!

Compositionality

- **Definition 4.27.** A meaning theory T is compositional, iff the meaning of an expression is a function of the meanings of its parts. We say that T obeys the compositionality principle or simply compositionality if it is.
- To compute the meaning of an expression, look up the meanings of the basic expressions forming it and successively compute the meanings of larger parts until a meaning for the whole expression is found.
- **Example 4.28 (Compositionality at work in arithmetic).** To compute the value of $(x+y)/(z \cdot u)$, look up the values of x, y, z, and u, then compute $x+y$ and $z \cdot u$, and finally compute the value of the whole expression.
- Many philosophers and linguists hold that compositionality is at work in ordinary language too.

Why Compositionality is Attractive

- Compositionality gives a nice building block for a meaning theory:
- **Example 4.29.** *[Expressions [are [built [from [words [that [combine [into [[larger [and larger]] subexpressions]]]]]]]]]*
- **Consequence:** To compute the meaning of an expression, look up the meanings of its words and successively compute the meanings of larger parts until a meaning for the whole expression is found.
- Compositionality explains how people can easily understand sentences they have never heard before, even though there are an infinite number of sentences any given person at any given time has not heard before.

Compositionality and the Congruence Principle

- Given reasonable assumptions compositionality entails the
- **Definition 4.30.** The congruence principle states that whenever A is part of B and A' means just the same as A, replacing A by A' in B will lead to a result that means just the same as B.
- **Example 4.31.** Consider the following (complex) sentences:
  1. *blah blah blah such and such blah blah*
  2. *blah blah blah so and so blah blah*

  If *such and such* and *so and so* mean the same thing, then 1. and 2. mean the same too.
- **Conversely:** if 1. and 2. do not mean the same, then *such and such* and *so and so* do not either.

A Test for Synonymity

- Suppose we accept the most certain principle (difference in truth conditions implies difference in meaning) and the congruence principle (replacing words by synonyms results in a synonymous utterance). Then we have a diagnostics for synonymy: Replacing utterances by synonyms preserves truth conditions, or equivalently
- **Definition 4.32.** The following is called the truth conditional synonymy test:

  > *If replacing A by B in some sentence C does not preserve truth conditions, then A and B are not synonymous.*

- We can use this as a test for the question of individuation: when are the meanings of two words the same – when are they synonymous?
- **Example 4.33 (Unsurprising Results).** The following sentences differ in truth conditions.
  1. *The cat is on the mat.*
  2. *The dog is on the mat.*

  Hence *cat* and *dog* are not synonymous. The converse holds for
  1. *John is a Greek.*
  2. *John is a Hellene.*

  In this case there is no difference in truth conditions.
- But there might be another context that does give a difference.

Contentious Cases of Synonymy Test

- **Example 4.34 (Problem).** The following sentences differ in truth values:
  1. *Mary believes that John is a Greek*
  2. *Mary believes that John is a Hellene*

  So *Greek* is not synonymous to *Hellene*. The same holds in the classical example:
  1. *The Ancients knew that Hesperus was Hesperus*
  2. *The Ancients knew that Hesperus was Phosphorus*

  In these cases most language users do perceive a difference in truth conditions while some philosophers vehemently deny that the sentences under 1. could be true in situations where the 2. sentences are false.
- It is important here of course that the context of substitution is within the scope of a verb of propositional attitude. (maybe later!)

A better Synonymy Test

- **Definition 4.35 (Synonymy).** The following is called the truth conditional synonymy test:

  If replacing A by B in some sentence C does not preserve truth conditions in a compositional part of C, then A and B are not synonymous.

Testing Truth Conditions with Logic

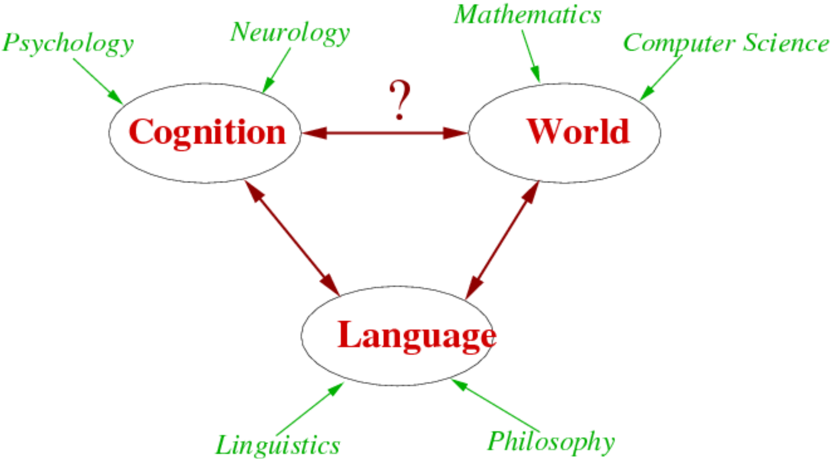
- **Definition 4.36.** A logical language model $\mathcal{M}$ for a natural language L consists of a logical system $\langle\mathcal{L},\mathcal{K},\models\rangle$ and a function φ from L sentences to $\mathcal{L}$-formulae.
- **Problem:** How do we find out whether $\mathcal{M}$ models L faithfully?
- **Idea:** Test truth conditions of sentences against the predictions $\mathcal{M}$ makes.
- **Problem:** The truth conditions for a sentence S in L can only be formulated and verified by humans that speak L.
- **In Practice:** Truth conditions are expressed as "stories" that specify salient situations. Native speakers of L are asked to judge whether they make S true/false.
- **Observation 4.37.** *A logical language model $\mathcal{M} := \langle L,\mathcal{L},\varphi\rangle$ can be tested:*
  1. *Select a sentence S and a situation W that makes S true in W.* (*according to humans*)
  2. *Translate S in to an $\mathcal{L}$-formula $S' := \varphi(S)$.*
  3. *Express W as a set Φ of $\mathcal{L}$-formulae.* ($\Phi \mathrel{\widehat{=}}$ *truth conditions*)
  4. *$\mathcal{M}$ is supported if $\Phi \models S'$, falsified if $\Phi \not\models S'$.*
- **Corollary 4.38.** *A logical language model constitutes a semantic meaning theory.*

2.5 Computational Semantics as a Natural Science

Computational Semantics as a Natural Science

- **In a nutshell:** Formal logic studies formal languages, their relation with the world (in particular the truth conditions). Computational logic adds the question about the computational behavior of the relevant aspects of the formal languages.
- This is almost the same as the task of natural language semantics!
- It is one of the key ideas that logics are good scientific models for natural languages, since they simplify certain aspects so that they can be studied in isolation. In particular, we can use the general scientific method of
  1. observing
  2. building formal theories for an aspect of reality,
  3. deriving the consequences of the hypotheses about the world in the theories
  4. testing the predictions made by the theory against the real-world data. If the theory predicts the data, then this supports the theory, if not, we refine the theory, starting the process again at 2.

NL Semantics as an Intersective Discipline



Part 1
English as a Formal Language: The Method of Fragments

Chapter 3
Logic as a Tool for Modeling NL Semantics

3.1 The Method of Fragments

Natural Language Fragments

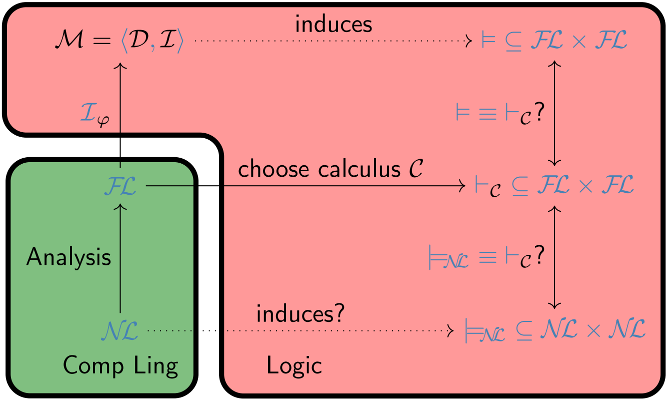
- **Methodological Problem:** How to organize the scientific method for natural language?
- **Delineation Problem:** What is natural language, e.g. English?
  Which aspects do we want to study?
- **Idea:** Select a subset (NL) sentences we want to study by a grammar!
  ⇝ Richard Montague's method of fragments (1972).
- **Definition 1.1.** The language L of a context-free grammar is called a fragment of a natural language N, iff $L \subseteq N$.
- **Scientific Fiction:** We can exhaust English with ever-increasing fragments, develop a semantic meaning theory for each.

Using CFGs for NL Fragments

- **Idea:** Use nonterminals to classify NL phrases.
- **Definition 1.2.** We call a nonterminal symbol of a context-free grammar a phrasal category. We distinguish two kinds of rules:
  - structural rules: $\mathcal{L}\colon H \to c_1, \ldots, c_n$ with head H, label $\mathcal{L}$, and a sequence of phrasal categories c_i.
  - lexical rules: $\mathcal{L}\colon H \to t_1 \mid \ldots \mid t_n$, where the t_i are terminals (i.e. NL phrases)
- **Definition 1.3.** In the method of fragments we use a CFG to parse sentences from the fragment into a parse tree (also called abstract syntax tree (AST) for further processing.
- **Todo:** We have to restrict our logical language models to fragments.
- **Definition 1.4.** A language fragment model consists of a CFG G, a logical system $\mathcal{L}$, and a semantics construction mapping φ from G-parse trees to $\mathcal{L}$-formulae.

Formal Natural Language Semantics with Fragments

- **Idea:** We will follow the picture we have discussed before



$$\begin{tikzcd}
\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle \arrow[rr, dotted, "\text{induces}"] & & \models \subseteq \mathcal{FL} \times \mathcal{FL} \arrow[d, leftrightarrow, "\models \equiv \vdash_{\mathcal{C}}?"'] \\
\mathcal{FL} \arrow[u, "\mathcal{I}_\varphi"] \arrow[rr, "\text{choose calculus } \mathcal{C}"] & & \vdash_{\mathcal{C}} \subseteq \mathcal{FL} \times \mathcal{FL} \arrow[d, leftrightarrow, "\models_{\mathcal{NL}} \equiv \vdash_{\mathcal{C}}?"'] \\
\mathcal{NL} \arrow[u, "\text{Analysis}"] \arrow[rr, dotted, "\text{induces?}"] & & \models_{\mathcal{NL}} \subseteq \mathcal{NL} \times \mathcal{NL}
\end{tikzcd}$$

Comp Ling

Logic

Choose a target logic $\mathcal{FL}$ and specify a translation from syntax trees to formulae!

Semantics by Translation

- **Idea:** We translate sentences by translating their syntax trees via tree node translation rules.
- **Note:** This makes the induced meaning theory compositional.
- **Definition 1.5.** We represent a node α in a syntax tree with children $\beta_1, \ldots, \beta_n$ by $[{X_1}_{\beta_1}, \ldots, {X_n}_{\beta_n}]_\alpha$ and write a translation rule as

  $$\mathcal{L}\colon [{X_1}_{\beta_1}, \ldots, {X_n}_{\beta_n}]_\alpha \leadsto \Phi({X_1}', \ldots, {X_n}')$$

  if the translation of the node α can be computed from those of the β_i via a semantical function Φ.
- **Definition 1.6.** For a natural language utterance or text A, we will use $\langle \mathrm{A} \rangle$ for the result of translating A and call it the interpretation of A.
- **Definition 1.7 (Default Rule).** For every word w in the fragment we assume a constant w' in the logic $\mathcal{L}$ and the "pseudo-rule" $t1\colon w \leadsto w'$. (if no other translation rule applies)

3.2 What is Logic?

What is Logic?

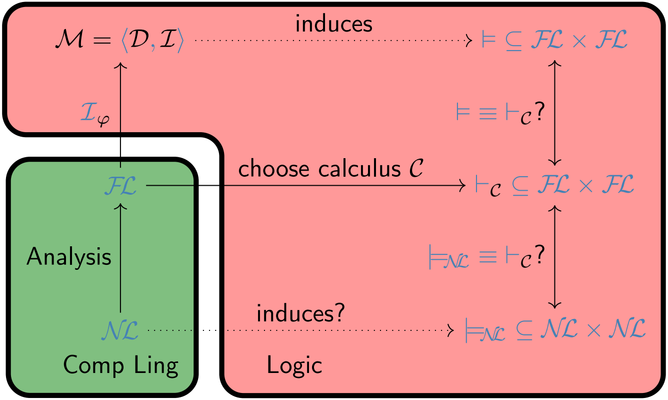
- **Definition 2.1.** Logic $\widehat{=}$ formal languages, inference and their relation with the world
  - Formal language $\mathcal{FL}$: set of formulae ($2+3/7$, $\forall x.x+y=y+x$)
  - Formula: sequence/tree of symbols ($x, y, f, g, p, 1, \pi, \in, \neg, \forall, \exists$)
  - Model: things we understand (e.g. number theory)
  - Interpretation: maps formulae into models ($[\![\text{three plus five}]\!]^{\mathcal{I}} = 8$)
  - Validity: $\mathcal{M} \models \mathrm{A}$, iff $[\![\mathrm{A}]\!]^{\mathcal{I}} = \mathrm{T}$ (five greater three is valid)
  - Entailment: $\mathrm{A} \models \mathrm{B}$, iff $\mathcal{M} \models \mathrm{B}$ for all $\mathcal{M} \models \mathrm{A}$. (generalize to $\mathcal{H} \models \mathrm{A}$)
  - Inference: rules to transform (sets of) formulae ($\mathrm{A}, \mathrm{A} \Rightarrow \mathrm{B} \vdash \mathrm{B}$)
  - Syntax: formulae, inference (just a bunch of symbols)
  - Semantics: models, interpr., validity, entailment (math. structures)
- **Important Question:** relation between syntax and semantics?

3.3 Using Logic to Model Meaning of Natural Language

Modeling Natural Language Semantics

- **Problem:** Find formal (logic) system for the meaning of natural language.
- History of ideas
  - Propositional logic [ancient Greeks like Aristotle]
    * *Every human is mortal*
  - First-Order Predicate logic [Frege $\leq$ 1900]
    * *I believe, that my audience already knows this.*
  - Modal logic [Lewis18, Kripke65]
    * *A man sleeps. He snores.* $((\exists X.\mathrm{man}(X) \wedge \mathrm{sleeps}(X))) \wedge \mathrm{snores}(X)$
  - Various dynamic approaches (e.g. DRT, DPL)
    * *Most men wear black*
  - Higher-order Logic, e.g. generalized quantifiers
  - ...

Natural Language Semantics?



$\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$

induces

$\models \subseteq \mathcal{FL} \times \mathcal{FL}$

$\mathcal{I}_\varphi$

$\models \equiv \vdash_\mathcal{C}$?

$\mathcal{FL}$

choose calculus $\mathcal{C}$

$\vdash_\mathcal{C} \subseteq \mathcal{FL} \times \mathcal{FL}$

Analysis

$\models_{\mathcal{NL}} \equiv \vdash_\mathcal{C}$?

$\mathcal{NL}$

induces?

$\models_{\mathcal{NL}} \subseteq \mathcal{NL} \times \mathcal{NL}$

Comp Ling

Logic

Logic-Based Knowledge Representation for NLP

- Logic (and related formalisms) allow to integrate world knowledge
  - explicitly (gives more understanding than statistical methods)
  - transparently (symbolic methods are monotonic)
  - systematically (we can prove theorems about our systems)
- Signal + world knowledge makes more powerful model
  - Does not preclude the use of statistical methods to guide inference
- Problems with logic-based approaches
  - Where does the world knowledge come from? (Ontology problem)
  - How to guide search induced by logical calculi? (combinatorial explosion)
- **One possible answer:** Description Logics. (Recall the AI-1 lecture?)

Chapter 4
Fragment 1

4.1 The First Fragment: Setting up the Basics

Fragment $\mathcal{F}_1$ Data (Sentences we want to cover)

- **Fragment $\mathcal{F}_1$ Data:** We delineate the intended fragment by giving examples
  1. *Ethel kicked the cat and Fiona laughted*
  2. *Peter is the teacher*
  3. *The teacher is happy*
  4. *It is not the case that Bertie ran*
  5. *It is not the case that Jo is happy*
- We can later use these sentences as benchmark tests.

4.1.1 Natural Language Syntax (Fragment 1)

Structural Grammar Rules

- **Definition 1.1.** $\mathcal{F}_1$ uses the following eight phrasal categories

S	sentence	NP	noun phrase
N	noun	N_{pr}	proper name
V^i	intransitive verb	V^t	transitive verb
conj	coordinator	Adj	adjective

- **Definition 1.2.** We have the following production rules in $\mathcal{F}_1$.
  $S1$: $S \rightarrow \mathrm{NP}\ V^i$,
  $S2$: $S \rightarrow \mathrm{NP}\ V^t\ \mathrm{NP}$,
  $N1$: $\mathrm{NP} \rightarrow N_{\mathrm{pr}}$,
  $N2$: $\mathrm{NP} \rightarrow$ the N,
  $S3$: $S \rightarrow$ It is not the case that S,
  $S4$: $S \rightarrow S\ \mathrm{conj}\ S$,
  $S5$: $S \rightarrow \mathrm{NP}$ is NP, and
  $S6$: $S \rightarrow \mathrm{NP}$ is Adj

Lexical insertion rules for Fragment $\mathcal{F}_1$

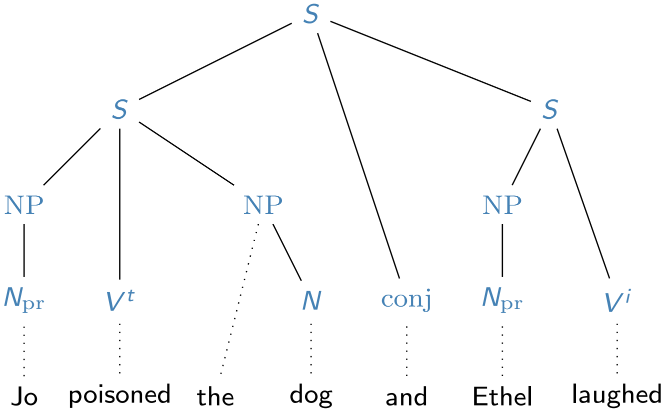
- **Definition 1.3.** We have the following lexical insertion rules in fragment $\mathcal{F}_1$.

$$\begin{aligned}
L1\colon & N_{\mathrm{pr}} \to \mathrm{Prudence} \mid \mathrm{Ethel} \mid \mathrm{Chester} \mid \mathrm{Jo} \mid \mathrm{Bertie} \mid \mathrm{Fiona},\\
L2\colon & N \to \mathrm{book} \mid \mathrm{cake} \mid \mathrm{cat} \mid \mathrm{golfer} \mid \mathrm{dog} \mid \mathrm{lecturer} \mid \mathrm{student} \mid \mathrm{singer},\\
L3\colon & V^i \to \mathrm{ran} \mid \mathrm{laughed} \mid \mathrm{sang} \mid \mathrm{howled} \mid \mathrm{screamed},\\
L4\colon & V^t \to \mathrm{read} \mid \mathrm{poisoned} \mid \mathrm{ate} \mid \mathrm{liked} \mid \mathrm{loathed} \mid \mathrm{kicked},\\
L5\colon & \mathrm{conj} \to \mathrm{and} \mid \mathrm{or},\\
L6\colon & \mathrm{Adj} \to \mathrm{happy} \mid \mathrm{crazy} \mid \mathrm{messy} \mid \mathrm{disgusting} \mid \mathrm{wealthy}
\end{aligned}$$

- **Definition 1.4.** A production rule whose head is a single non-terminal and whose body consists of a single terminal is called lexical or a lexical insertion rule.
- **Notation:** Lexical insertion rules are usually written using BNF alternative in the body ⇜ grouping rules with the same head.
- **Definition 1.5.** The subset of lexical rules of a grammar G is called the lexicon of G and the set of body symbols the vocabulary (or alphabet). The nonterminals in their heads are called lexical categories of G.
- **Note:** We will adopt the convention that new lexical insertion rules can be generated spontaneously as needed.

Syntax Example: *Jo poisoned the dog and Ethel laughed*

- **Observation 1.6.** *Jo poisoned the dog and Ethel laughed is a sentence of fragment 1*
- We can construct a parse tree for it!



4.1.2 Predicate Logic without Quantifiers

Individuals and their Properties/Relationships

- **Observation:** We want to talk about individuals like Stefan, Nicole, and Jochen
  and their properties, e.g. being blond, or studying AI
  and relationships, e.g. that *Stefan loves Nicole*.
- **Idea:** Re-use PL^0, but replace propositional variables with something more expressive! (instead of fancy variable name trick)

Individuals and their Properties/Relationships

- **Observation:** We want to talk about individuals like Stefan, Nicole, and Jochen
  and their properties, e.g. being blond, or studying AI
  and relationships, e.g. that *Stefan loves Nicole*.
- **Idea:** Re-use PL^0, but replace propositional variables with something more expressive! (instead of fancy variable name trick)
- **Definition 1.8.** A first-order signature $\langle\Sigma^f, \Sigma^p\rangle$ consists of
  - $\Sigma^f := \bigcup_{k\in\mathbb{N}}\Sigma^f_k$ of function constants, where members of Σ^f_k denote k-ary functions on individuals,
  - $\Sigma^p := \bigcup_{k\in\mathbb{N}}{\Sigma^p}_k$ of predicate constants, where members of ${\Sigma^p}_k$ denote k-ary relations among individuals,

  where Σ^f_k and ${\Sigma^p}_k$ are pairwise disjoint, countable sets of symbols for each $k \in \mathbb{N}$.
  A 0-ary function constant refers to a single individual, therefore we call it a individual constant.

A Grammar for $\mathrm{PL}^{\mathrm{nq}}$

- **Definition 1.9.** The formulae of $\mathrm{PL}^{\mathrm{nq}}$ are given by the following grammar

function constants	f^k	$\in$	Σ_k^f	
predicate constants	p^k	$\in$	$\Sigma^p{}_k$	
terms	t	$::=$	f^0	individualconstant
		$\mid$	$f^k(t_1, \ldots, t_k)$	application
formulae	A	$::=$	$p^k(t_1, \ldots, t_k)$	atomic
		$\mid$	$\neg \mathrm{A}$	negation
		$\mid$	$\mathrm{A}_1 \wedge \mathrm{A}_2$	conjunction

$\mathrm{PL}^{\mathrm{nq}}$ Semantics

- **Definition 1.10.** Domains $\mathcal{D}_0 = \{\mathsf{T}, \mathsf{F}\}$ of truth values and $\mathcal{D}_\iota \neq \emptyset$ of individuals.
- **Definition 1.11.** Interpretation $\mathcal{I}$ assigns values to constants, e.g.
  - $\mathcal{I}(\neg)\colon \mathcal{D}_0 \to \mathcal{D}_0; \mathsf{T} \mapsto \mathsf{F}; \mathsf{F} \mapsto \mathsf{T}$ and $\mathcal{I}(\wedge) = \ldots$ (as in PL^0)
  - $\mathcal{I}\colon \Sigma_0^f \to \mathcal{D}_\iota$ (interpret individual constants as individuals)
  - $\mathcal{I}\colon \Sigma_k^f \to {\mathcal{D}_\iota}^k \to \mathcal{D}_\iota$ (interpret function constants as functions)
  - $\mathcal{I}\colon {\Sigma^p}_k \to \mathcal{P}({\mathcal{D}_\iota}^k)$ (interpret predicate constants as relations)
- **Definition 1.12.** The value function $\mathcal{I}$ assigns values to formulae: (recursively)
  - $\mathcal{I}(f(\mathrm{A}^1, \ldots, \mathrm{A}^k)) := \mathcal{I}(f)(\mathcal{I}(\mathrm{A}^1), \ldots, \mathcal{I}(\mathrm{A}^k))$
  - $\mathcal{I}(p(\mathrm{A}^1, \ldots, \mathrm{A}^k)) := \mathsf{T}$, iff $\langle \mathcal{I}(\mathrm{A}^1), \ldots, \mathcal{I}(\mathrm{A}^k) \rangle \in \mathcal{I}(p)$
  - $\mathcal{I}(\neg \mathrm{A}) = \mathcal{I}(\neg)(\mathcal{I}(\mathrm{A}))$ and $\mathcal{I}(\mathrm{A} \wedge \mathrm{B}) = \mathcal{I}(\wedge)(\mathcal{I}(\mathrm{A}), \mathcal{I}(\mathrm{G}))$ (just as in PL^0)
- **Definition 1.13.** Model: $\mathcal{M} = \langle \mathcal{D}_\iota, \mathcal{I} \rangle$ varies in $\mathcal{D}_\iota$ and $\mathcal{I}$.
- **Theorem 1.14.** $\mathrm{PL}^{\mathrm{nq}}$ *is isomorphic to* PL^0 *(interpret atoms as prop. variables)*

A Model for $\mathrm{PL}^{\mathrm{nq}}$

- **Example 1.15.** Let $L := \{a, b, c, d, e, P, Q, R, S\}$, we set the universe $\mathcal{D} := \{♣, ♠, ♡, ♢\}$, and specify the interpretation function $\mathcal{I}$ by setting
  - $a \mapsto ♣$, $b \mapsto ♠$, $c \mapsto ♡$, $d \mapsto ♢$, and $e \mapsto ♢$ for constants,
  - $P \mapsto \{♣, ♠\}$ and $Q \mapsto \{♠, ♢\}$, for unary predicate constants.
  - $R \mapsto \{\langle ♡, ♢\rangle, \langle ♢, ♡\rangle\}$, and $S \mapsto \{\langle ♢, ♠\rangle, \langle ♠, ♣\rangle\}$ for binary predicate constants.
- **Example 1.16 (Computing Meaning in this Model).**
  - $\mathcal{I}(R(a, b) \wedge P(c)) = \mathsf{T}$, iff
  - $\mathcal{I}(R(a, b)) = \mathsf{T}$ and $\mathcal{I}(P(c)) = \mathsf{T}$, iff
  - $\langle \mathcal{I}(a), \mathcal{I}(b)\rangle \in \mathcal{I}(R)$ and $\mathcal{I}(c) \in \mathcal{I}(P)$, iff
  - $\langle ♣, ♠\rangle \in \{\langle ♡, ♢\rangle, \langle ♢, ♡\rangle\}$ and $♡ \in \{♣, ♠\}$

  So, $\mathcal{I}(R(a, b) \wedge P(c)) = \mathsf{F}$.

$\mathrm{PL}^{\mathrm{nq}}$ and PL^0 are Isomorphic

- **Observation:** For every choice of Σ of signature, the set $\mathcal{A}_\Sigma$ of atomic $\mathrm{PL}^{\mathrm{nq}}$ formulae is countable, so there is a $\mathcal{V}_\Sigma \subseteq \mathcal{V}_0$ and a bijection $\theta_\Sigma\colon \mathcal{A}_\Sigma \to \mathcal{V}_\Sigma$. θ_Σ can be extended to formulae as $\mathrm{PL}^{\mathrm{nq}}$ and PL^0 share connectives.
- **Lemma 1.17.** *For every model $\mathcal{M} = \langle \mathcal{D}_\iota, \mathcal{I} \rangle$, there is a variable assignment $\varphi_\mathcal{M}$, such that $\mathcal{I}_{\varphi_\mathcal{M}}(\mathsf{A}) = \mathcal{I}(\mathsf{A})$.*
- *Proof sketch:* We just define $\varphi_\mathcal{M}(X) := \mathcal{I}(\theta_\Sigma^{-1}(X))$
- **Lemma 1.18.** *For every variable assignment $\psi\colon \mathcal{V}_\Sigma \to \{\mathsf{T},\mathsf{F}\}$ there is a model $\mathcal{M}^\psi = \langle \mathcal{D}^\psi, \mathcal{I}^\psi \rangle$, such that $\mathcal{I}_\psi(\mathsf{A}) = \mathcal{I}^\psi(\mathsf{A})$.*
- *Proof sketch:* see next slide
- **Corollary 1.19.** *$\mathrm{PL}^{\mathrm{nq}}$ is isomorphic to PL^0, i.e. the following diagram commutes:*

$$\begin{tikzcd}
\langle \mathcal{D}^\psi, \mathcal{I}^\psi \rangle & \mathcal{V}_\Sigma \to \{\mathsf{T},\mathsf{F}\} \arrow[l, "\psi \mapsto \mathcal{M}^\psi"'] \\
\mathrm{PL}^{\mathrm{nq}}(\Sigma) \arrow[u, "\mathcal{I}^\psi()"] \arrow[r, "\theta_\Sigma"] & \mathrm{PL}^0(\mathcal{A}_\Sigma) \arrow[u, "\mathcal{I}_{\varphi_\mathcal{M}}()"']
\end{tikzcd}$$

- **Note:** This constellation with a language isomorphism and a corresponding model isomorphism (in converse direction) is typical for a logic isomorphism.

Valuation and Satisfiability

- **Lemma 1.20.** *For every variable assignment $\psi\colon \mathcal{V}_\Sigma \to \{\mathsf{T},\mathsf{F}\}$ there is a model $\mathcal{M}^\psi = \langle \mathcal{D}^\psi, \mathcal{I}^\psi \rangle$, such that $\mathcal{I}_\psi(\mathrm{A}) = \mathcal{I}^\psi(\mathrm{A})$.*
- *Proof:* We construct $\mathcal{M}^\psi = \langle \mathcal{D}^\psi, \mathcal{I}^\psi \rangle$ and show that it works as desired.
  1. Let $\mathcal{D}^\psi$ be the set of $\mathrm{PL}^{\mathrm{nq}}$ terms over Σ, and
     - $\mathcal{I}^\psi(f) :\ {\mathcal{D}_\iota}^k \to {\mathcal{D}^\psi}^k ; \langle \mathrm{A}_1, \ldots, \mathrm{A}_k \rangle \mapsto f(\mathrm{A}_1, \ldots, \mathrm{A}_k)$ for $f \in \Sigma^f_k$
     - $\mathcal{I}^\psi(p) := \{\langle \mathrm{A}_1, \ldots, \mathrm{A}_k \rangle \mid \psi(\theta_\psi^{-1} p(\mathrm{A}_1, \ldots, \mathrm{A}_k)) = \mathsf{T}\}$ for $p \in \Sigma^p$.
  2. We show $\mathcal{I}^\psi(\mathrm{A}) = \mathrm{A}$ for terms A by induction on A
     - 2.1. If $\mathrm{A} = c$, then $\mathcal{I}^\psi(\mathrm{A}) = \mathcal{I}^\psi(c) = c = \mathrm{A}$
     - 2.2. If $\mathrm{A} = f(\mathrm{A}_1, \ldots, \mathrm{A}_n)$ then $\mathcal{I}^\psi(\mathrm{A}) = \mathcal{I}^\psi(f)(\mathcal{I}(\mathrm{A}_1), \ldots, \mathcal{I}(\mathrm{A}_n)) = \mathcal{I}^\psi(f)(\mathrm{A}_1, \ldots, \mathrm{A}_k) = \mathrm{A}$.
  3. For a $\mathrm{PL}^{\mathrm{nq}}$ formula A we show that $\mathcal{I}^\psi(\mathrm{A}) = \mathcal{I}_\psi(\mathrm{A})$ by induction on A.
     - 3.1. If $\mathrm{A} = p(\mathrm{A}_1, \ldots, \mathrm{A}_k)$, then $\mathcal{I}^\psi(\mathrm{A}) = \mathcal{I}^\psi(p)(\mathcal{I}(\mathrm{A}_1), \ldots, \mathcal{I}(\mathrm{A}_n)) = \mathsf{T}$, iff $\langle \mathrm{A}_1, \ldots, \mathrm{A}_k \rangle \in \mathcal{I}^\psi(p)$, iff $\psi(\theta_\psi^{-1}\mathrm{A}) = \mathsf{T}$, so $\mathcal{I}^\psi(\mathrm{A}) = \mathcal{I}_\psi(\mathrm{A})$ as desired.
     - 3.2. If $\mathrm{A} = \neg \mathrm{B}$, then $\mathcal{I}^\psi(\mathrm{A}) = \mathsf{T}$, iff $\mathcal{I}^\psi(\mathrm{B}) = \mathsf{F}$, iff $\mathcal{I}^\psi(\mathrm{B}) = \mathcal{I}_\psi(\mathrm{B})$, iff $\mathcal{I}^\psi(\mathrm{A}) = \mathcal{I}_\psi(\mathrm{A})$.
     - 3.3. If $\mathrm{A} = \mathrm{B} \wedge \mathrm{C}$ then we argue similarly
  4. Hence $\mathcal{I}^\psi(\mathrm{A}) = \mathcal{I}_\psi(\mathrm{A})$ for all $\mathrm{PL}^{\mathrm{nq}}$ formulae and we have concluded the proof.

4.1.3 Natural Language Semantics via Translation

Translation rules for non-basic expressions (NP and S)

- **Definition 1.21.** We have the following translation rules for non-leaf node of the syntax tree
  $T1$: $[X_{\mathrm{NP}}, Y_{V^i}]_S \rightsquigarrow Y'(X')$
  $T2$: $[X_{\mathrm{NP}}, Y_{V^t}, Z_{\mathrm{NP}}]_S \rightsquigarrow Y'(X', Z')$
  $T3$: $[X_{N_{\mathrm{pr}}}]_{\mathrm{NP}} \rightsquigarrow X'$
  $T4$: $[\mathrm{the}, X_N]_{\mathrm{NP}} \rightsquigarrow theX'$
  $T5$: $[\text{It is not the case that} X_S]_S \rightsquigarrow (\neg X')$
  $T6$: $[X_S, Y_{\mathrm{conj}}, Z_S]_S \rightsquigarrow Y'(X', Z')$
  $T7$: $[X_{\mathrm{NP}}, \mathrm{is}, Y_{\mathrm{NP}}]_S \rightsquigarrow X' = Y'$
  $T8$: $[X_{\mathrm{NP}}, \mathrm{is}, Y_{\mathrm{Adj}}]_S \rightsquigarrow Y'(X')$
  Read e.g. $[Y, Z]_X$ as a node with label X in the syntax tree with children X and Y. Read X' as the translation of X via these rules.
- Note that we have exactly one translation per syntax rule.

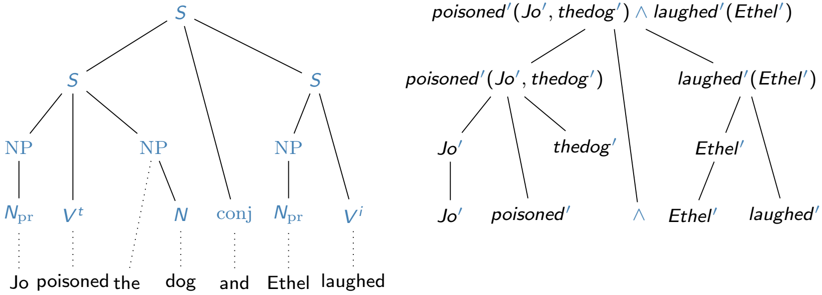
Translation rule for basic lexical items

- **Definition 1.22.** The target logic for $\mathcal{F}_1$ is $\mathrm{PL}^{\mathrm{nq}}$, the fragment of PL^1 without quantifiers.
- **Lexical Translation Rules for $\mathcal{F}_1$ Categories:**
  - If w is a proper name, then $w' \in \Sigma_0^f$. (individual constant)
  - If w is an intransitive verb, then $w' \in \Sigma^p{}_1$. (one-place predicate)
  - If w is a transitive verb, $w' \in \Sigma^p{}_2$. (two-place predicate)
  - If w is a noun phrase, then $w' \in \Sigma_0^f$. (individual constant)
- **Semantics by Translation:** We translate sentences by translating their syntax trees via tree node translation rules.
- For any lexical item (i.e. word) w, we have the "pseudo-rule" $t1\colon w \rightsquigarrow w'$.
- **Note:** This rule does not apply to the syncategorematic items is and the.
- Translations for logical connectives

$$t2\colon \text{and} \rightsquigarrow \wedge,\ t3\colon \text{or} \rightsquigarrow \vee,\ t4\colon \text{it is not the case that} \rightsquigarrow \neg$$

Translation Example

- **Observation 1.23.** *Jo poisoned the dog and Ethel laughed is a sentence of fragment $\mathcal{F}_1$.*
- We can construct a syntax tree for it!



4.2 Testing Truth Conditions via Inference

Testing Truth Conditions in $\mathrm{PL}^{\mathrm{nq}}$

- **Idea 1:** To test our language model ($\mathcal{F}_1$)
  - Select a sentence S and a situation W that makes S true. (according to humans)
  - Translate S in to a formula S' in $\mathrm{PL}^{\mathrm{nq}}$.
  - Express W as a set Φ of formulae in $\mathrm{PL}^{\mathrm{nq}}$ ($\Phi \widehat{=}$ truth conditions)
  - Our language model is supported if $\Phi \models S'$, falsified if $\Phi \not\models S'$.
- **Example 2.1 (John chased the gangster in the red sports car).**
  - We claimed that we have three readings **??**
    $R_1 := c(j,g) \wedge in(j,s)$, $R_2 := c(j,g) \wedge in(g,s)$, and $R_3 := c(j,g) \wedge in(j,s) \wedge in(g,s)$
  - So there must be three distinct situations W that make S true
    1. *John is in the red sports car, but the gangster isn't*
       $W_1 := c(j,g) \wedge in(j,s) \wedge \neg in(g,s)$, so $W_1 \models R_1$, but $W_1 \not\models R_2$ and $W_1 \not\models R_3$
    2. *The gangster is in the red sports car, but John isn't*
       $W_2 := c(j,g) \wedge in(g,s) \wedge \neg in(j,s)$, so $W_2 \models R_2$, but $W_2 \not\models R_1$ and $W_2 \not\models R_3$
    3. *Both are in the red sports car*
       $\widehat{=}$ they run around on the back seat of a very big sports car
       $W_3 := c(j,g) \wedge in(j,s) \wedge in(g,s)$, so $W_3 \models R_3$, but $W_3 \not\models R_1$ and $W_3 \not\models R_1$
- **Idea 2:** Use a calculus to model $\models$, e.g. $\mathcal{ND}_0$

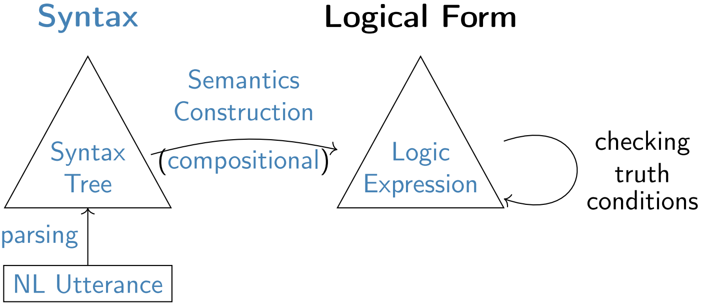
4.3 Summary & Evaluation

Fragment $\mathcal{F}_1$ – Summary

- Fragment $\mathcal{F}_1$ of English (defined by grammar + lexicon)
- Logic $\mathrm{PL^{nq}}$ (serves as a mathematical model for $\mathcal{F}_1$)
  - Formal Language (individuals, predicates, $\neg, \wedge, \vee, \Rightarrow$)
  - Semantics $\mathcal{I}_\varphi$ defined recursively on formula structure ($\rightsquigarrow$ validity, entailment)
  - Tableau calculus for validity and entailment (CALCULEMUS!)
- Analysis function $\mathcal{F}_1 \rightsquigarrow \mathrm{PL^{nq}}$ (Translation)
- Test the model by checking predictions (calculate truth conditions)
- **Coverage:** Extremely Boring! (accounts for 0 examples from the intro) but the conceptual setup is fascinating

Summary: The Interpretation Process (so far)

- **The Interpretation Process in $\mathcal{F}_1$:** Can be visualized in the following diagram:



Chapter 5
Fragment 2: Pronouns and World Knowledge $\leadsto$ Semantic/Pragmatic Analysis

5.1 Fragment 2: Pronouns and Anaphora

Fragment $\mathcal{F}_2$ ($\mathcal{F}_2 \widehat{=} \mathcal{F}_1$ + Anaphoric Pronouns)

- **Want to cover:** *Peter loves Fido. He bites him.* (almost intro)
  - **We need**: Translation and interpretation for pronouns like *he*, *she*, *him*,. . . .
  - **Also**: A way to integrate world knowledge to filter out one interpretation. (i.e. *Humans don't bite dogs.*)
- **Idea:** Integrate variables into $\mathrm{PL^{nq}}$ (work backwards from that)
- **Logical System:** $\mathrm{PL^{nq}}(\mathcal{V}) = \mathrm{PL^{nq}} +$ variables (Translate pronouns to variables)

New Grammar in $\mathcal{F}_2$ (Pronouns)

- **Definition 1.1.** We have the following structural grammar rules in $\mathcal{F}_2$

$$
\begin{aligned}
S1\colon & S \rightarrow \mathrm{NP}, V^i,\\
S2\colon & S \rightarrow \mathrm{NP}, V^t, \mathrm{NP},\\
N1\colon & \mathrm{NP} \rightarrow N_{\mathrm{pr}},\\
N2\colon & \mathrm{NP} \rightarrow \mathrm{Pron},\\
N3\colon & \mathrm{NP} \rightarrow \text{the}, N,\\
S3\colon & S \rightarrow \text{it is not the case that}, S,\\
S4\colon & S \rightarrow S, \mathrm{conj}, S,\\
S5\colon & S \rightarrow \mathrm{NP}, \text{is}, \mathrm{NP},\\
S6\colon & S \rightarrow \mathrm{NP}, \text{is}, \mathrm{Adj}
\end{aligned}
$$

and one additional lexical rule:

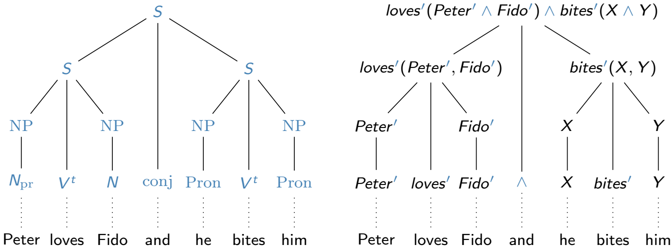
$$L7\colon \mathrm{Pron} \rightarrow he \mid she \mid it \mid we \mid they$$

Predicate Logic with Variables (but no Quantifiers)

- **Definition 1.2 (Logical System $\mathrm{PL}^{\mathrm{nq}}(\mathcal{V})$).** $\mathrm{PL}^{\mathrm{nq}}(\mathcal{V}) := \mathrm{PL}^{\mathrm{nq}} +$ variables
- **Definition 1.3 ($\mathrm{PL}^{\mathrm{nq}}(\mathcal{V})$ Syntax).**
  Category $\mathcal{V} = \{X, Y, Z, X^1, X^2, \ldots\}$ of variables (allow variables wherever individual constants were allowed)
- **Definition 1.4 ($\mathrm{PL}^{\mathrm{nq}}(\mathcal{V})$ Semantics).**
  First-order model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ (need to evaluate variables)
  - variable assignment: $\varphi \colon \mathcal{V}_\iota \to U$
  - value function: $\mathcal{I}_\varphi(X) = \varphi(X)$ (defined like $\mathcal{I}$ elsewhere)
  - call a $\mathrm{PL}^{\mathrm{nq}}(\mathcal{V})$ formula A valid in $\mathcal{M}$ under φ, iff $\mathcal{I}_\varphi(\mathrm{A}) = \mathsf{T}$,
  - call it satisfiable in $\mathcal{M}$, iff there is a variable assignment φ, such that $\mathcal{I}_\varphi(\mathrm{A}) = \mathsf{T}$

Translation for $\mathcal{F}_2$ (first attempt)

- **Idea:** Pronouns are translated into new variables (so far)
- **New Translation Rule:** We translate pronouns by the "rule":
  $T9$: $[X]_{\mathrm{Pron}} \rightsquigarrow Y_{new}$, where Y_{new} is a new variable.
- The syntax/semantic trees for *Peter loves Fido and he bites him.* are straightforward. (almost intro)



5.2 Inference with World Knowledge and Free Variables – A Case Study

5.2.1 Pragmatics via Model Generation Tableaux?

A Tableau Calculus for $\mathrm{PL}^{\mathrm{nq}}(\mathcal{V})$

- **Definition 2.1 (Tableau Calculus for $\mathrm{PL}^{\mathrm{nq}}(\mathcal{V})$).** $\mathcal{T}_{\mathcal{V}}^{p} = \mathcal{T}_0$ + new tableau rules for formulae with variables

$$\frac{\begin{array}{c}\vdots\\ \mathrm{A}^{\alpha}\\ \vdots\end{array}\quad c\in\mathcal{H}}{([c/X](\mathrm{A}))^{\alpha}}\ \mathcal{T}_{\mathcal{V}}^{p}WK \qquad\qquad \frac{\begin{array}{c}\vdots\\ \boxed{\mathrm{A}}\end{array}\quad \begin{array}{c}\mathcal{H}=\{a_1,\ldots,a_n\}\\ \mathrm{free}(\mathrm{A})=\{X_1,\ldots,X_m\}\end{array}}{(\sigma_1(\mathrm{A}))^{\mathsf{T}}\ \Big|\ \ldots\ \Big|\ (\sigma_{n^m}(\mathrm{A}))^{\mathsf{T}}}\ \mathcal{T}_{\mathcal{V}}^{p}Ana$$

$\mathcal{H}$ is the set of ind. constants in the branch above (Herbrand universe) and the σ_i are substitutions that instantiate the X_j with any combinations of the a_k (there are n^m of them).

- the first rule is used for world knowledge (up in the branch)
- the second rule is used for input logical forms $\boxed{\cdots}$

  this rule has to be applied eagerly (while they are still at the leaf)

To allow for world knowledge, we generalize the notion of an initial tableau. Instead of allowing only the initial labeled formula at the root node, we allow a linear tree whose nodes are labeled formulae with positive formulae representing the world knowledge. As the world knowledge resides in the initial tableau (intuitively before all input), we will also speak of background knowledge.

Some Examples in $\mathcal{F}_2$

- **Example 2.2 (Peter snores).** (Only sleeping people snore)

$$\begin{array}{c}
(\mathrm{snores}(X) \Rightarrow \mathrm{sleeps}(X))^{\mathsf{T}} \\
\boxed{\mathrm{snores}(\mathrm{peter})} \\
(\mathrm{snores}(\mathrm{peter}) \Rightarrow \mathrm{sleeps}(\mathrm{peter}))^{\mathsf{T}} \\
\mathrm{sleeps}(\mathrm{peter})^{\mathsf{T}}
\end{array}$$

- **Example 2.3 (Peter sleeps. John walks. He snores).** (who snores?)

$$\begin{array}{c}
\boxed{\mathrm{sleeps}(\mathrm{peter})} \\
\boxed{\mathrm{walks}(\mathrm{john})} \\
\boxed{\mathrm{snores}(X)} \\
\begin{array}{c|c}
\mathrm{snores}(\mathrm{peter})^{\mathsf{T}} & \mathrm{snores}(\mathrm{john})^{\mathsf{T}}
\end{array}
\end{array}$$

Does Tweety Fly? The everlasting Question in AI

- **Example 2.4.**

Tweety is a bird

$$\begin{array}{c} (\mathrm{bird}(X) \Rightarrow (\mathrm{flies}(X) \vee \mathrm{penguin}(X)))^{\mathsf{T}} \\ (\mathrm{penguin}(X) \Rightarrow \neg\mathrm{flies}(X))^{\mathsf{T}} \\ \boxed{\mathrm{bird}(\mathrm{tweety})} \\ (\mathrm{flies}(\mathrm{tweety}) \vee \mathrm{penguin}(\mathrm{tweety}))^{\mathsf{T}} \\ \begin{array}{c|c} \mathrm{flies}(\mathrm{tweety})^{\mathsf{T}} & \mathrm{penguin}(\mathrm{tweety})^{\mathsf{T}} \\ & \neg\mathrm{flies}(\mathrm{tweety})^{\mathsf{T}} \\ & \mathrm{flies}(\mathrm{tweety})^{\mathsf{F}} \end{array} \end{array}$$

Tweety is an eagle

$$\begin{array}{c} (\mathrm{bird}(X) \Rightarrow (\mathrm{flies}(X) \vee \mathrm{penguin}(X) \\ (\mathrm{eagle}(X) \Rightarrow \mathrm{bird}(X))^{\mathsf{T}} \\ (\mathrm{penguin}(X) \Rightarrow \neg\mathrm{eagle}(X))^{\mathsf{T}} \\ (\mathrm{penguin}(X) \Rightarrow \neg\mathrm{flies}(X))^{\mathsf{T}} \\ \boxed{\mathrm{eagle}(\mathrm{tweety})} \\ \mathrm{bird}(\mathrm{tweety})^{\mathsf{T}} \\ (\mathrm{flies}(\mathrm{tweety}) \vee \mathrm{penguin}(\mathrm{tweety})) \\ \begin{array}{c|c} \mathrm{flies}(\mathrm{tweety})^{\mathsf{T}} & \mathrm{penguin}(\mathrm{tweety})^{\mathsf{T}} \\ & (\neg\mathrm{eagle}(\mathrm{tweety})) \\ & \mathrm{eagle}(\mathrm{tweety})^{\mathsf{F}} \\ & \bot \end{array} \end{array}$$

- For the second we need to add more world knowledge.

5.2.2 Case Study: Peter loves Fido, even though he sometimes bites him

Finally: *Peter loves Fido. He bites him.*

- Let's try it naively (worry about the problems later.)

$$\begin{array}{c}\boxed{l(p,f)}\\ \boxed{b(X,Y)}\\ b(p,p)^{\top} \;\big|\; b(p,f)^{\top} \;\big|\; b(f,p)^{\top} \;\big|\; b(f,f)^{\top}\end{array}$$

- **Problem:** We get four readings instead of one!
- **Idea:** We have not specified enough world knowledge.

Peter and Fido with World Knowledge

- Nobody bites himself, humans do not bite dogs.

$$
\begin{array}{c}
d(f)^{\mathsf{T}} \\
m(p)^{\mathsf{T}} \\
b(X,X)^{\mathsf{F}} \\
(d(X) \wedge m(Y) \Rightarrow \neg b(Y,X))^{\mathsf{T}} \\
\boxed{l(p,f)} \\
\boxed{b(X,Y)} \\
\begin{array}{c|c|c|c}
\begin{array}{c} b(p,p)^{\mathsf{T}} \\ b(p,p)^{\mathsf{F}} \\ \bot \end{array} &
\begin{array}{c} b(p,f)^{\mathsf{T}} \\ (d(f) \wedge m(p) \Rightarrow \neg b(p,f))^{\mathsf{T}} \\ b(p,f)^{\mathsf{F}} \\ \bot \end{array} &
\begin{array}{c} b(f,p)^{\mathsf{T}} \\ \\ \\ \\ \end{array} &
\begin{array}{c} b(f,f)^{\mathsf{T}} \\ b(f,f)^{\mathsf{F}} \\ \bot \end{array}
\end{array}
\end{array}
$$

- **Observation:** Anaphor resolution introduces ambiguities.
- **Pragmatics:** Use world knowledge to filter out impossible readings.

5.2.3 The Computational Role of Ambiguities

The computational Role of Ambiguities

- **Observation:** (in the traditional waterfall model)
  Every processing stage introduces ambiguities that need to be resolved.
  - Syntax: e.g. *Peter chased the man in the red sports car* (attachment)
  - Semantics: e.g. *Peter went to the bank* (lexical)
  - Pragmatics: e.g. *Two men carried two bags* (collective vs. distributive)
- **Question:** Where does pronoun ambiguity belong? (much less clear)
- **Answer:** we have freedom to choose
  1. resolve the pronouns in the syntax (generic waterfall model)
     - ⇝ multiple syntactic representations (pragmatics as filter)
  2. resolve the pronouns in the pragmatics (our model here)
     - ⇝ need underspecified syntactic representations (e.g. variables)
     - ⇝ pragmatics needs ambiguity treatment (e.g. tableaux)

Translation for Fragment $\mathcal{F}_2$ Reconsidered

- **Idea:** Pronouns are translated into *new variables*. (so far)
- **Problem:** *Peter loves Mary. She loves him.*

$$\boxed{\mathrm{loves}(\mathrm{peter}, \mathrm{mary})}$$
$$\boxed{\mathrm{loves}(X, Y)}$$
$$\mathrm{loves}(\mathrm{peter}, \mathrm{peter})^{\mathsf{T}} \mid \mathrm{loves}(\mathrm{peter}, \mathrm{mary})^{\mathsf{T}} \mid \mathrm{loves}(\mathrm{mary}, \mathrm{peter})^{\mathsf{T}} \mid \mathrm{loves}(\mathrm{mary}, \mathrm{mary})^{\mathsf{T}}$$

- **Idea:** Attach world knowledge to pronouns. (just as with Peter and Fido)
  - Use the world knowledge to distinguish (linguistic) gender by predicates masc and fem.
- **Problem:** Properties of
  - proper names are given in the model,
  - pronouns must be given by the syntax-semantics interface.
- **In particular:** How to generate $\mathrm{loves}(X, Y) \wedge \mathrm{masc}(X) \wedge \mathrm{fem}(Y)$ compositionally?

Sorts refine World Categories

- **Definition 2.5 (Sorted Logics).** (in our case $PL^1_{\mathcal{S}}$)
  Assume a set of sorts $\mathcal{S} := \{\mathbb{A}, \mathbb{B}, \mathbb{C}, \ldots\}$, annotate every syntactic and semantic structure with them. Make all constructions and operations well worted:
  - Syntax: Variables and constants are sorted $X_{\mathbb{A}}, Y_{\mathbb{B}}, Z^1_{\mathbb{C}_1} \ldots, a_{\mathbb{A}}, b_{\mathbb{A}}, \ldots$
  - Semantics: Subdivide the universe $\mathcal{D}$ into subsets $\mathcal{D}_{\mathbb{A}} \subseteq \mathcal{D}$
    Interpretation $\mathcal{I}$ and variable assignment φ have to be well-sorted.
    $\mathcal{I}(a_{\mathbb{A}}), \varphi(X_{\mathbb{A}}) \in \mathcal{D}_{\mathbb{A}}$.
  - Calculus: Substitutions must be well sorted $[a_{\mathbb{A}}/X_{\mathbb{A}}]$ OK, $[a_{\mathbb{A}}/X_{\mathbb{B}}]$ not.
- **Observation:** Sorts do not add expressivity in principle (just practically) For every sort $\mathbb{A}$, we introduce a first-order predicate $\mathcal{R}_{\mathbb{A}}$ and
  - Translate $R(X_{\mathbb{A}}) \wedge \neg P(Z_{\mathbb{C}})$ to $\mathcal{R}_{\mathbb{A}}(X) \wedge \mathcal{R}_{\mathbb{C}}(Z) \Rightarrow R(X) \wedge \neg P(Z)$ in world knowledge.
  - Translate $R(X_{\mathbb{A}}) \wedge \neg P(Z_{\mathbb{C}})$ to $\mathcal{R}_{\mathbb{A}}(X) \wedge \mathcal{R}_{\mathbb{C}}(Z) \wedge R(X, Y) \wedge \neg P(Z)$ in input.
  - Meaning is preserved, but translation is non-compositional!

5.3 Tableaux and Model Generation

5.3.1 Tableau Branches and Herbrand Models

Model Generation and Interpretation

- **Example 3.1 (from above).** In **??** we claimed that the set

$$\mathcal{B} := \{\mathrm{loves}(\mathrm{john},\mathrm{mary})^{\mathsf{F}}, \mathrm{loves}(\mathrm{mary},\mathrm{bill})^{\mathsf{T}}\}$$

  of literals on the open branch of the tableau $\mathcal{T}$ below

$$\begin{array}{c|c} \multicolumn{2}{c}{(\mathrm{loves}(\mathrm{mary},\mathrm{bill}) \vee \mathrm{loves}(\mathrm{john},\mathrm{mary}))^{\mathsf{T}}} \\ \multicolumn{2}{c}{\mathrm{loves}(\mathrm{john},\mathrm{mary})^{\mathsf{F}}} \\ \mathrm{loves}(\mathrm{mary},\mathrm{bill})^{\mathsf{T}} & \mathrm{loves}(\mathrm{john},\mathrm{mary})^{\mathsf{T}} \\ & \bot \end{array}$$

  constitutes a “model”. (it can be conveniently read off)

- **Recap:** A first-order model $\mathcal{M}$ is a pair $\langle\mathcal{D},\mathcal{I}\rangle$, where $\mathcal{D}$ is a set of individuals, and $\mathcal{I}$ is an interpretation function.
- **Problem:** Find $\mathcal{D}$ and $\mathcal{I}$ based on $\mathcal{B}$.

Model Generation and Models

- **Idea:** Choose the universe $\mathcal{D}$ as the set Σ_0^f of constants, choose $\mathcal{I} = \mathrm{Id}_{\Sigma_0^f}$, interpret $p \in {\Sigma^p}_k$ as $\mathcal{R}_{\mathcal{B}}(p) := \{\langle a_1, \ldots, a_k\rangle \mid p(a_1, \ldots, a_k)^{\mathsf{T}} \in \mathcal{B}\}$.
- **Definition 3.2.** We call a model a Herbrand model, iff $\mathcal{D} = \Sigma_0^f$ and $\mathcal{I} = \mathrm{Id}_{\Sigma_0^f}$.
- **Definition 3.3.** Let $\mathcal{H}$ be a set of atomic propositions such that $\mathsf{A}^{\mathsf{F}} \notin \mathcal{H}$, if $\mathsf{A}^{\mathsf{T}} \in \mathcal{H}$, then we call $\mathcal{H}$ a Herbrand valuation.
- **Lemma 3.4.** *Let $\mathcal{H}$ be a Herbrand valuation, then setting $\mathcal{I}(p) := \mathcal{R}_{\mathcal{H}}(p)$ yields a Herbrand model that satisfies $\mathcal{H}$.* (proof trivial)
- **Corollary 3.5.** *Let $\mathcal{H}$ be a Herbrand valuation, then there is a Herbrand model that satisfies $\mathcal{H}$.* (use $\mathcal{R}_{\mathcal{H}}$)

5.3.2 Using Model Generation for Interpretation

Using Model Generation for Interpretation

- **Definition 3.6.** Mental model theory [JL83; JLB91] posits that humans form mental models of the world, i.e. (neural) representations of possible states of the world that are consistent with the perceptions up to date and use them to reason about the world.
- **So** communication by natural language is a process of transporting parts of the mental model of the speaker into the mental model of the hearer.
- **Therefore** the NL interpretation process on the part of the hearer is a process of integrating the meaning of the utterances of the speaker into his mental model.
- **Idea:** We can model discourse understanding as a process of generating Herbrand models for the logical form of an utterance in a discourse by a tableau based model generation procedure.
- **Advantage:** Capturing ambiguity by generating multiple models for input logical forms.

Tableau Machine

- **Definition 3.7.** The tableau machine is an inferential cognitive model for incremental natural language understanding that implements mental model theory via tableau based model generation over a sequence of input sentences. It iterates the following process for every input sentence staring with the empty tableau:

  1. add the logical form of the input sentence S_i to the selected branch,
  2. perform tableau inferences below S_i until saturated or some resource criterion is met
  3. if there are open branches choose a "preferred branch", otherwise backtrack to previous tableau for S_j with $j < i$ and open branches, then re-process $S_{j+1}, \ldots, S_i$ if possible, else fail.

  The output is application-dependent; some choices are

  - the Herbrand model for the preferred branch $\leadsto$ preferred interpretation;
  - the literals augmented with all non-expanded formulae (from the discourse); (resource-bound was reached)
  - Tableau machine answers user queries (preferred model $\models$ query?)

- Interpretation mode via model generation (guided by resources and strategies)
- Query mode by refutation theorem proving (□ for side conditions; using tableau rules)

The Tableau Machine in Model Generation Mode

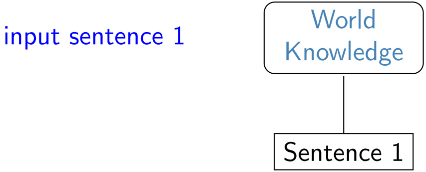
- **Example 3.8.** The tableau machine in action (query mode on two sentences).

initialize tableau

World Knowledge

The Tableau Machine in Model Generation Mode

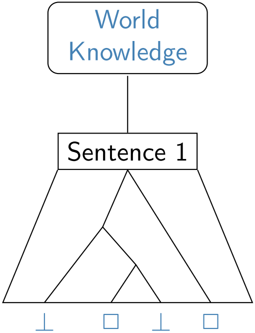
- **Example 3.9.** The tableau machine in action (query mode on two sentences).



The Tableau Machine in Model Generation Mode

- **Example 3.10.** The tableau machine in action (query mode on two sentences).

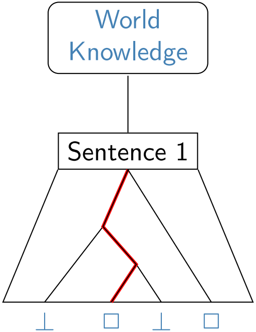
saturate tableau



The Tableau Machine in Model Generation Mode

- **Example 3.11.** The tableau machine in action (query mode on two sentences).

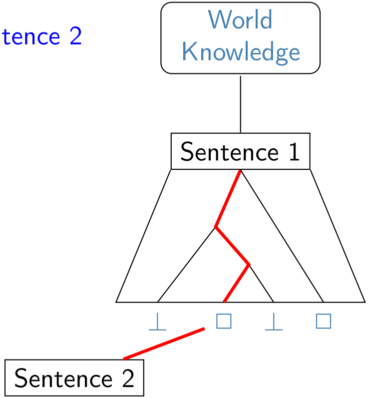
choose branch



The Tableau Machine in Model Generation Mode

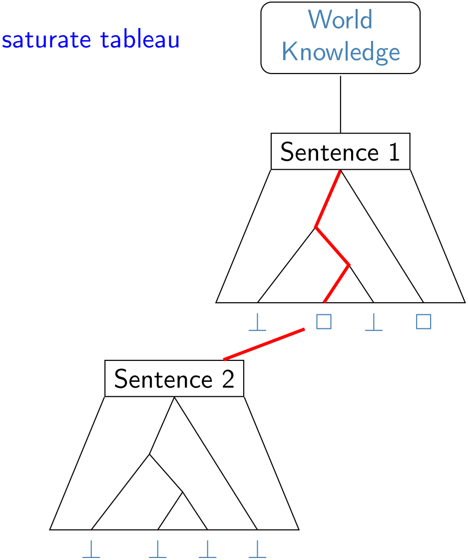
- **Example 3.12.** The tableau machine in action (query mode on two sentences).

input sentence 2



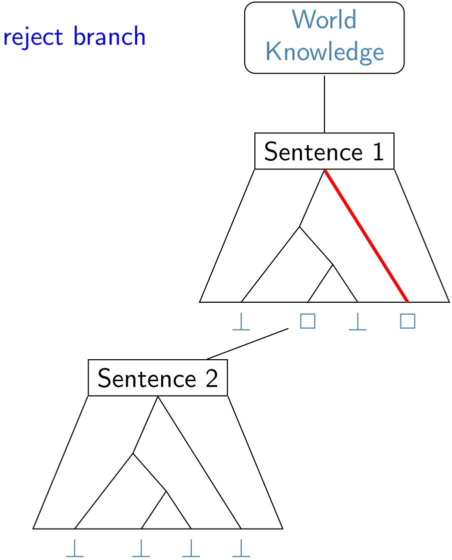
The Tableau Machine in Model Generation Mode

- **Example 3.13.** The tableau machine in action (query mode on two sentences).



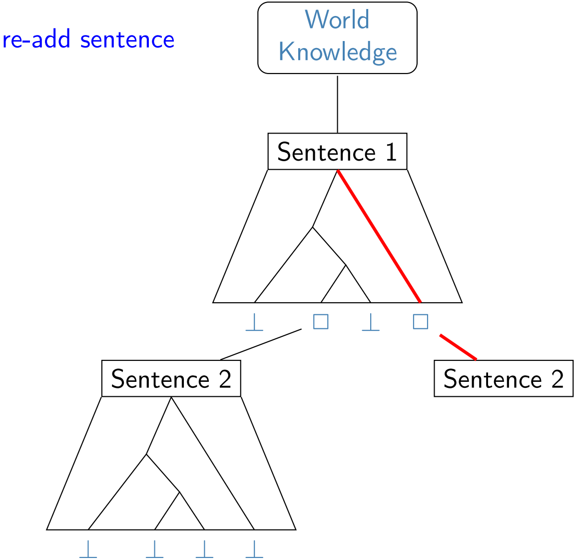
The Tableau Machine in Model Generation Mode

- **Example 3.14.** The tableau machine in action (query mode on two sentences).



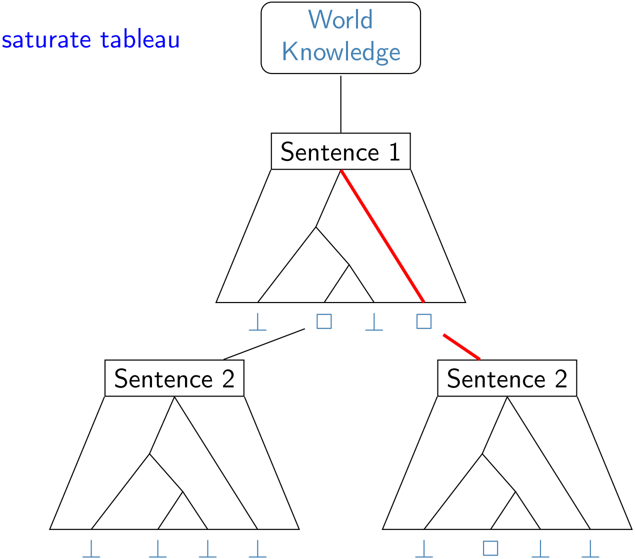
The Tableau Machine in Model Generation Mode

- **Example 3.15.** The tableau machine in action (query mode on two sentences).



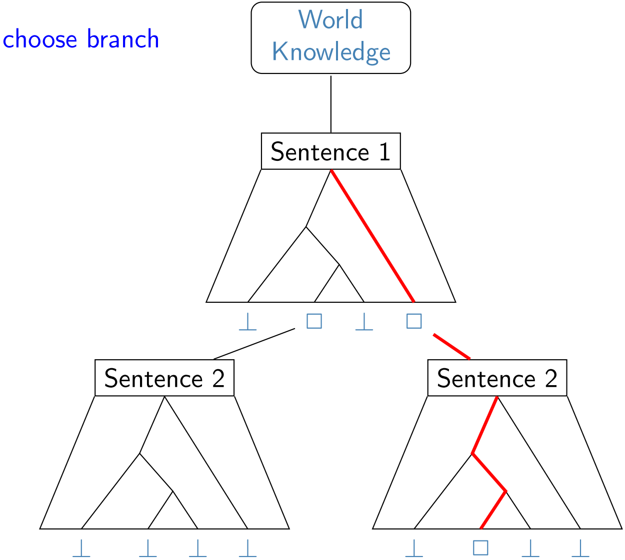
The Tableau Machine in Model Generation Mode

- **Example 3.16.** The tableau machine in action (query mode on two sentences).



The Tableau Machine in Model Generation Mode

- **Example 3.17.** The tableau machine in action (query mode on two sentences).



Two (Syntactical) Readings

- **Example 3.18 (A syntactically ambiguous sentence).**
  Peter loves Mary and Mary sleeps or Peter snores.

  Reading 1: $\mathrm{loves}(\mathrm{peter},\mathrm{mary})\wedge(\mathrm{sleeps}(\mathrm{mary})\vee\mathrm{snores}(\mathrm{peter}))$

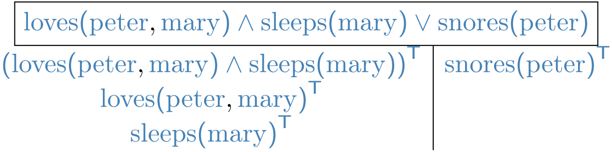
  Reading 2: $\mathrm{loves}(\mathrm{peter},\mathrm{mary})\wedge\mathrm{sleeps}(\mathrm{mary})\vee\mathrm{snores}(\mathrm{peter})$

  Consider the first reading, start out with the empty tableau for simplicity, even though this is cognitively implausible.

$$\begin{array}{c}\boxed{\mathrm{loves}(\mathrm{peter},\mathrm{mary})\wedge(\mathrm{sleeps}(\mathrm{mary})\vee\mathrm{snores}(\mathrm{peter}))}\\ \mathrm{loves}(\mathrm{peter},\mathrm{mary})^{\mathsf{T}}\\ (\mathrm{sleeps}(\mathrm{mary})\vee\mathrm{snores}(\mathrm{peter}))^{\mathsf{T}}\\ \mathrm{sleeps}(\mathrm{mary})^{\mathsf{T}}\ \Big|\ \mathrm{snores}(\mathrm{peter})^{\mathsf{T}}\end{array}$$

- **Observation:** We have two models, so we have a case of pragmatic ambiguity.

The other (Syntactical) Reading



$$\boxed{\mathrm{loves}(\mathrm{peter},\mathrm{mary}) \wedge \mathrm{sleeps}(\mathrm{mary}) \vee \mathrm{snores}(\mathrm{peter})}$$

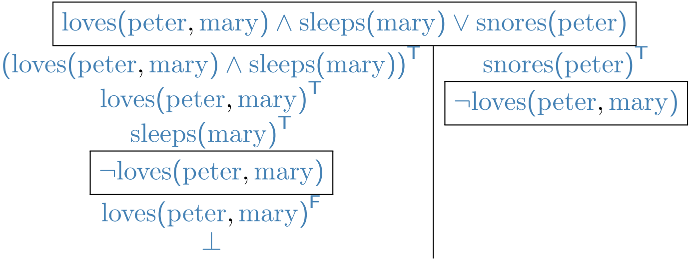
$$(\mathrm{loves}(\mathrm{peter},\mathrm{mary}) \wedge \mathrm{sleeps}(\mathrm{mary}))^{\mathsf{T}} \quad \Big| \quad \mathrm{snores}(\mathrm{peter})^{\mathsf{T}}$$

$$\mathrm{loves}(\mathrm{peter},\mathrm{mary})^{\mathsf{T}}$$

$$\mathrm{sleeps}(\mathrm{mary})^{\mathsf{T}}$$

Continuing the Discourse

- **Example 3.19.** *Peter does not love Mary*.
  Then the second tableau would be extended to



$\boxed{\mathrm{loves}(\mathrm{peter},\mathrm{mary}) \wedge \mathrm{sleeps}(\mathrm{mary}) \vee \mathrm{snores}(\mathrm{peter})}$

$(\mathrm{loves}(\mathrm{peter},\mathrm{mary}) \wedge \mathrm{sleeps}(\mathrm{mary}))^{\mathsf{T}}$ | $\mathrm{snores}(\mathrm{peter})^{\mathsf{T}}$

$\mathrm{loves}(\mathrm{peter},\mathrm{mary})^{\mathsf{T}}$ | $\boxed{\neg\mathrm{loves}(\mathrm{peter},\mathrm{mary})}$

$\mathrm{sleeps}(\mathrm{mary})^{\mathsf{T}}$

$\boxed{\neg\mathrm{loves}(\mathrm{peter},\mathrm{mary})}$

$\mathrm{loves}(\mathrm{peter},\mathrm{mary})^{\mathsf{F}}$

$\bot$

  and the first tableau closes altogether.

- In effect the choice of models has been reduced to one, which constitutes the intuitively correct reading of the discourse.

5.3.3 Adding Equality to PLNQ for Fragment 1

$\mathrm{PL}^{=}_{\mathrm{NQ}}$: Adding Equality to $\mathrm{PL}^{\mathrm{nq}}$

- **Syntax:** Just another binary predicate constant $=$
- **Semantics:** Fixed as $\mathcal{I}_\varphi(a = b) = \mathsf{T}$, iff $\mathcal{I}_\varphi(a) = \mathcal{I}_\varphi(b)$. (logical constant)
- **Definition 3.20 (Tableau Calculus $\mathcal{T}^{=}_{\mathrm{NQ}}$).** Add two additional inference rules (a positive and a negative) to $\mathcal{T}_0$

$$\frac{a \in \mathcal{H}}{a = a^{\mathsf{T}}}\;\mathcal{T}^{=}_{\mathrm{NQ}}\mathrm{sym} \qquad\qquad \frac{\begin{array}{c} a = b^{\mathsf{T}} \\ \mathrm{A}\,[a]_p{}^{\alpha} \end{array}}{[b/p]\mathrm{A}^{\alpha}}\;\mathcal{T}^{=}_{\mathrm{NQ}}\mathrm{rep}$$

where

- $\mathcal{H} \mathrel{\widehat{=}}$ the Herbrand universe, i.e. the set of constants occurring on the branch.
- we write $\mathrm{C}\,[\mathrm{A}]_p$ to indicate that $\mathrm{C}|_p = \mathrm{A}$ (C has subterm A at position p).
- $[\mathrm{A}/p]\mathrm{C}$ is obtained from C by replacing the subterm at position p with A.

- **Note:** We could have equivalently written $\mathcal{T}^{=}_{\mathrm{NQ}}\mathrm{sym}$ as $\dfrac{a = a^{\mathsf{F}}}{\bot}$:

  With $\mathcal{T}^{=}_{\mathrm{NQ}}\mathrm{sym}$ conjure $a = a^{\mathsf{T}}$ from thin air, use it to close $a = a^{\mathsf{F}}$.
- **So, . . .** $\mathcal{T}^{=}_{\mathrm{NQ}}\mathrm{sym}$ and $\mathcal{T}^{=}_{\mathrm{NQ}}\mathrm{rep}$ follow the pattern of having a T and a F rule per logical constant.

Reading Comprehension Example: Mini TOEFL test

- **Example 3.21 (Reading Comprehension).** If you hear/read *Mary is the teacher. Peter likes the teacher.*, do you know whether *Peter likes Mary*?
- **Idea:** Interpret via tableau machine (interpretation mode) and test entailment in query mode.

Reading Comprehension Example: Mini TOEFL test

- **Example 3.22 (Reading Comprehension).** If you hear/read *Mary is the teacher. Peter likes the teacher.*, do you know whether *Peter likes Mary*?
- **Idea:** Interpret via tableau machine (interpretation mode) and test entailment in query mode.
- **Interpretation:** Feed $\Phi_1 := \mathrm{mary} = \mathrm{the\_teacher}$ and $\Phi_2 := \mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})$ to the tableau machine in turn. Model generation tableau (nothing to do on these inputs)

$$\boxed{\mathrm{mary} = \mathrm{the\_teacher}}$$
$$\boxed{\mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})}$$

Reading Comprehension Example: Mini TOEFL test

- **Example 3.23 (Reading Comprehension).** If you hear/read *Mary is the teacher. Peter likes the teacher.*, do you know whether *Peter likes Mary*?
- **Idea:** Interpret via tableau machine (interpretation mode) and test entailment in query mode.
- **Interpretation:** Feed $\Phi_1 := \mathrm{mary} = \mathrm{the\_teacher}$ and $\Phi_2 := \mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})$ to the tableau machine in turn.
- **Question Answering:** Use the tableau machine in query mode for an "entailment test": Label $\varphi := \mathrm{likes}(\mathrm{peter}, \mathrm{mary})$ with F and saturate.

$$\begin{array}{c}
\boxed{\mathrm{mary} = \mathrm{the\_teacher}}\\
\boxed{\mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})}\\
\left\Vert\begin{array}{c}
\mathrm{likes}(\mathrm{peter}, \mathrm{mary})^{\mathsf{F}}\\
\mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})^{\mathsf{F}}\\
\bot
\end{array}\right\Vert
\end{array}$$

Indeed, it closes, so $\Phi_1, \Phi_2 \models \varphi \rightsquigarrow$ *yes, Peter likes Mary*.

Reading Comprehension Example: Mini TOEFL test

- **Example 3.24 (Reading Comprehension).** If you hear/read *Mary is the teacher. Peter likes the teacher.*, do you know whether *Peter likes Mary*?
- **Idea:** Interpret via tableau machine (interpretation mode) and test entailment in query mode.
- **Interpretation:** Feed $\Phi_1 := \mathrm{mary} = \mathrm{the\_teacher}$ and $\Phi_2 := \mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})$ to the tableau machine in turn.
- **Question Answering:** Use the tableau machine in query mode for an "entailment test": Label $\varphi := \mathrm{likes}(\mathrm{peter}, \mathrm{mary})$ with F and saturate.

  $$\boxed{\mathrm{mary} = \mathrm{the\_teacher}}$$
  $$\boxed{\mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})}$$
  $$\left\|\begin{array}{c} \mathrm{likes}(\mathrm{peter}, \mathrm{mary})^{\mathsf{F}} \\ \mathrm{likes}(\mathrm{peter}, \mathrm{the\_teacher})^{\mathsf{F}} \\ \bot \end{array}\right\|$$

  Indeed, it closes, so $\Phi_1, \Phi_2 \models \varphi \leadsto$ *yes, Peter likes Mary*.
- **Note:** The part marked in double vertical lines is removed from the tableau after answering. (do not mess up the tree/models)

5.4 Summary & Evaluation

Fragment $\mathcal{F}_2$ – Summary

- Fragment $\mathcal{F}_2$ extends $\mathcal{F}_1$ by pronouns.
- Logic/translation extended correspondingly:
  - Equality (actually already needed for $\mathcal{F}_1$)
  - Variables as underspecified representations for anaphoric pronouns.
- New NLU component: semantic/pragmatic analysis
  - Tableau machine as an inferential model for pronoun resolution.
  - Uses world knowledge to augment/prune models.
- **Coverage:** Still relatively limited (accounts for 1 example from the intro)

Model Generation models Discourse Understanding

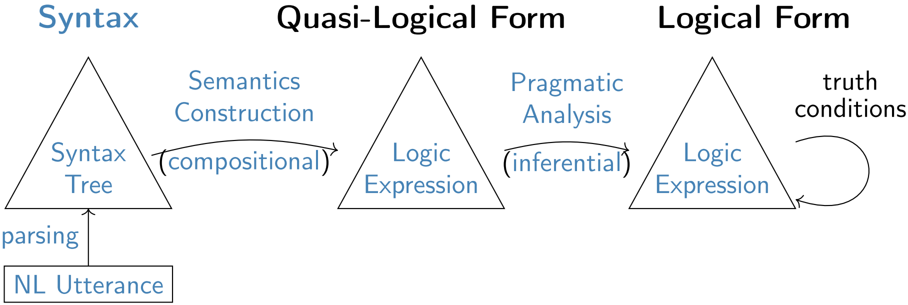
- The tableau machine algorithm conforms with psycholinguistic findings:
  - Zwaan& Radvansky [ZR98]: listeners not only represent logical form, but also models containing referents.
  - deVega [de 95]: online, incremental process.
  - Singer [Sin94]: enriched by background knowledge.
  - Glenberg et al. [GML87]: major function is to provide basis for anaphor resolution.

Towards a Performance Model for NLU

- **Problem:** The tableau machine is only a competence model.
- **Definition 4.1.** A competence model is a meaning theory that delineates a space of possible discourses. A performance model delineates the discourses actually used in communication. (after [Cho65])
- **Idea:** We need to guide the tableau machine in which inferences and branch choices it performs.
- **Idea:** Each tableau rule comes with rule costs.
  - **Here**: each sentence in the discourse has a fixed inference budget. Expansion until budget used up.
  - **Ultimately** we want bounded optimization regime [Rus91]: Expansion as long as expected gain in model quality outweighs proof costs
- **Effect:** Expensive rules are rarely applied. (only if the promise great rewards)
- ⚠ Finding appropriate values for rule costs and model quality is an open problem.

Summary: The Full Interpretation Process

- **Full Interpretation Process:** In $\mathcal{F}_2$ we have extended the interpretation process by semantic/pragmatic analysis, so we arrive at:



Chapter 6
Fragment 3: Complex Verb Phrases

6.1 Fragment 3 (Handling Verb Phrases)

$\mathcal{F}_3$: New Data (Verb Phrases)

- New Data: in $\mathcal{F}_3$.
  1. *Ethel howled and screamed.*
  2. *Ethel kicked the dog and poisoned the cat.*
  3. *Fiona liked Jo and loathed Ethel and tolerated Prudence.*
  4. *Fiona kicked the cat and laughed.*
  5. *Bertie didn't laugh.*
  6. *Bertie didn't laugh and didn't scream.*
  7. *Bertie didn't laugh or scream.*
  8. *Bertie didn't laugh or kick the dog.*
  9. * *Bertie didn't didn't laugh.*
- We extend $\mathcal{F}_2$. (no feature interaction)

New Grammar in Fragment $\mathcal{F}_3$ (Verb Phrases)

- To account for the syntax we come up with the concept of a verb phrase (VP)
- **Definition 1.1.** A verb phrase is any phrase that can be used (syntactially) whereever a verb can be.
- **Example 1.2.** The phrase *tolerated Prudence* is like *slept* (syntactially)
- **Idea:** Allow verb phrases (VP in the grammar wherever we had intransitive verbs (V^i) before.
- **Problem:** The obvious rule $VP \rightarrow$ didn't VP over-generates: it accepts * *Bertie didn't didn't laugh.* (note the infinitive)
- **Definition 1.3.** A verb is called finite, iff it contextually complements either an explicit subject or – in the imperative mood – an implicit subject.
- **Observation:** Finite verbs are inflected.
- **Definition 1.4.** Non-finite verbs, are verb forms that do not show tense, person, or number.
- **Idea:** We will use features +fin for finite, -fin for non-finite in grammar rules, and ±fin for schemata.

New Grammar in Fragment $\mathcal{F}_3$ (Verb Phrases)

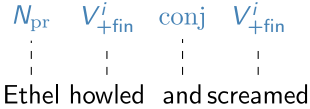
- **Definition 1.5.** $\mathcal{F}_3$ has the following rules:

S1.	S	!:	→	NP $VP_{+\text{fin}}$
S2.	S	!:	→	S conj S
V1.	$VP_{\pm\text{fin}}$	!:	→	$V^i_{\pm\text{fin}}$
V2.	$VP_{\pm\text{fin}}$	!:	→	$V^t_{\pm\text{fin}}$ NP
V3.	$VP_{\pm\text{fin}}$	!:	→	$VP_{\pm\text{fin}}$ conj $VP_{\pm\text{fin}}$
V4.	$VP_{+\text{fin}}$	!:	→	$BE_{=}$ NP
V5.	$VP_{+\text{fin}}$	!:	→	BE_{pred} Adj.
V6.	$VP_{+\text{fin}}$	!:	→	didn't $VP_{-\text{fin}}$

N1.	NP	→	N_{pr}
N2.	NP	→	Pron
N3.	NP	→	the N
L8.	$BE_{=}$	→	is
L9.	BE_{pred}	→	is
L10.	$V^i_{-\text{fin}}$	→	run, laugh,...
L11.	$V^t_{-\text{fin}}$	→	read, poison,...

- **Remark:** The ±fin feature solves the "didn't" over-generation problem.
- **Remark:** Many machine-oriented grammars have extensive feature systems like our ±fin.
- **Limitations of $\mathcal{F}_3$:**
  - $\mathcal{F}_3$ does not allow coordination of transitive verbs (problematic anyways)
    Prudence kicked and scratched Ethel.

Testing the Grammar on an Example

- **Example 1.6.** 

Ethel howled and screamed

Towards a Semantics for $\mathcal{F}_3$

- **Recall:** So far we have mapped intransitive verb (V^i) to predicates which could be applied to NP meanings (individuals).
- **So:** *VP* meanings are functions from individuals to truth values
- **And:** conj meanings are functionals that map functions to functions.
- In logic we distinguish such objects (individuals and functions of various kinds) by assigning them types.
- Let's make this formal $\leadsto$ develop a suitable logic!

6.2 Dealing with Functions in Logic and Language

Types

- **Intuition:** Types are semantic annotations for terms that prevent antinomies.
- **Definition 2.1.** Given a set $\mathcal{BT}$ of base types, construct function types: $\alpha \to \beta$ is the type of functions with domain type α and range type β. We call the closure $\mathcal{T}$ of $\mathcal{BT}$ under function types the set of simple types over $\mathcal{BT}$.
- **Definition 2.2.** We will use ι for the type of individuals and o for the type of truth values.
- **Right Associativity:** The type constructor is used as a right-associative operator, i.e. we use $\alpha \to \beta \to \gamma$ as an abbreviation for $\alpha \to (\beta \to \gamma)$
- **Vector Notation:** We will use a kind of vector notation for function types, abbreviating $\alpha_1 \to \ldots \to \alpha_n \to \beta$ with $\overline{\alpha}_n \to \beta$.

What can happen without Types as a Safety-Net

- **Definition 2.3.** The unrestricted comprehension principle states that for any sufficiently well-defined property P, there is the set of all and only the objects that have property P.
- **Definition 2.4.** Russell's paradox (also known as Russell's antinomy) is a set-theoretic paradox that shows that every set theory that contains an unrestricted comprehension principle leads to contradictions.
- **Definition 2.5.** The Russell set R is the set of all sets that are not members of themselves.

What can happen without Types as a Safety-Net

- **Definition 2.6.** The unrestricted comprehension principle states that for any sufficiently well-defined property P, there is the set of all and only the objects that have property P.
- **Definition 2.7.** Russell's paradox (also known as Russell's antinomy) is a set-theoretic paradox that shows that every set theory that contains an unrestricted comprehension principle leads to contradictions.
- **Definition 2.8.** The Russell set R is the set of all sets that are not members of themselves.
- **Observation:** If R is assumed to exist (e.g. by the unrestricted comprehension principle), then we end up with an antinomy:
  - Suppose $R \in R$, then then we must have $R \notin R$, since we have explicitly taken out the set that contain themselves.
  - Suppose $R \notin R$, then have $R \in R$, since all other sets are elements.

  So $R \in R$ iff $R \notin R$, which is a contradiction! (Russell's Antinomy [Rus03])

What can happen without Types as a Safety-Net

- **Definition 2.9.** The unrestricted comprehension principle states that for any sufficiently well-defined property P, there is the set of all and only the objects that have property P.
- **Definition 2.10.** Russell's paradox (also known as Russell's antinomy) is a set-theoretic paradox that shows that every set theory that contains an unrestricted comprehension principle leads to contradictions.
- **Definition 2.11.** The Russell set R is the set of all sets that are not members of themselves.
- **Observation:** If R is assumed to exist (e.g. by the unrestricted comprehension principle), then we end up with an antinomy:
  - Suppose $R \in R$, then then we must have $R \notin R$, since we have explicitly taken out the set that contain themselves.
  - Suppose $R \notin R$, then have $R \in R$, since all other sets are elements.

  So $R \in R$ iff $R \notin R$, which is a contradiction! (Russell's Antinomy [Rus03])
- **Does Logic help?:**
  - No, if untyped: $R := \{m \mid m \notin m\}$ or equivalently: $R := \{m \mid m\, m\}$.
  - Yes, if typed: $m(m)$ cannot be well-typed with simple types, so we can not define R.
- **Generally:** Simple types prevent self-application: If we type $m(m)$ as $m_\alpha(m_\beta)$, then we must have $\alpha = \beta \to \gamma$ for the function application to work but also $\alpha = \beta$ to have consistent typing.

Syntactical Categories and Types

- Now, we can assign types to syntactic categories.

Cat	Type	Intuition
S	o	truth value
NP	ι	individual
N_{pr}	ι	individuals
VP	$\iota \to o$	property
V^i	$\iota \to o$	unary predicate
V^t	$\iota \to \iota \to o$	binary relation

- For the category conj, we cannot get by with a single type. Depending on where it is used, we need the types
  - $o \to o \to o$ for S-coordination in rule $S2$: $S \to S\ \mathrm{conj}\ S$
  - $(\iota \to o) \to (\iota \to o) \to (\iota \to o)$ for VP-coordination in $V3$: $VP \to VP\ \mathrm{conj}\ VP$.
- **Note:** Computational Linguistics, often uses a different notation for types: e (entity) for ι, t (truth value) for o, and $\langle\alpha,\beta\rangle$ for $\alpha \to \beta$ (no bracket elision convention).
  So the type for VP-coordination has the form $\langle\langle e,t\rangle,\langle\langle e,t\rangle,\langle e,t\rangle\rangle\rangle$

From Comprehension to β-Conversion

- $\exists F_{\alpha\to\beta}.\forall X_\alpha.FX = \mathrm{A}_\beta$ for arbitrary variable X_α and term $\mathrm{A} \in \mathit{wff}_\beta(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ (for each term A and each variable X there is a function $f \in \mathcal{D}_{\alpha\to\beta}$, with $f(\varphi(X)) = \mathcal{I}_\varphi(\mathrm{A})$)
  - schematic in α, β, X_α and A_β, very inconvenient for deduction
- Transformation in $\mathcal{H}_\Omega$
  - $\exists F_{\alpha\to\beta}.\forall X_\alpha.FX = \mathrm{A}_\beta$
  - $\forall X_\alpha.(\lambda X_\alpha.\mathrm{A})X = \mathrm{A}_\beta$ ($\exists E$)
    Call the function F whose existence is guaranteed "$(\lambda X_\alpha.\mathrm{A})$"
  - $(\lambda X_\alpha.\mathrm{A})\mathrm{B} = [\mathrm{B}/X]\mathrm{A}_\beta$ ($\forall E$), in particular for $\mathrm{B} \in \mathit{wff}_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$.
- **Definition 2.12.** Axiom of β equality: $(\lambda X_\alpha.\mathrm{A})\,\mathrm{B} = [\mathrm{B}/X](\mathrm{A}_\beta)$
- **Idea:** Introduce a new class of formulae (λ-calculus [Chu40])

From Extensionality to η-Conversion

- **Definition 2.13.** Extensionality Axiom: $\forall F_{\alpha\to\beta}.\forall G_{\alpha\to\beta}.(\forall X_\alpha.FX = GX) \Rightarrow F = G$
- **Idea:** Maybe we can get by with a simplified equality schema here as well.
- **Definition 2.14.** We say that A and $\lambda X_\alpha.\mathrm{A}\; X$ are η-equal, (write $\mathrm{A}_{\alpha\to\beta} =_\eta \lambda X_\alpha.\mathrm{A}\; X$), iff $X \notin \mathrm{free}(\mathrm{A})$.
- **Theorem 2.15.** *η-equality and Extensionality are equivalent*
- *Proof:* We show that η-equality is special case of extensionality; the converse direction is trivial
  1. Let $\forall X_\alpha.\mathrm{A}X = \mathrm{B}X$, thus $\mathrm{A}X = \mathrm{B}X$ with $\forall E$
  2. $\lambda X_\alpha.\mathrm{A}X = \lambda X_\alpha.\mathrm{B}X$, therefore $\mathrm{A} = \mathrm{B}$ with η
  3. Hence $\forall F_{\alpha\to\beta}.\forall G_{\alpha\to\beta}.(\forall X_\alpha.FX = GX) \Rightarrow F = G$ by twice $\forall I$.
- Axiom of truth values: $\forall F_o.\forall G_o.FG \Leftrightarrow F = G$ unsolved.

6.3 Simply Typed λ-Calculus

Simply typed λ-Calculus (Syntax)

- **Definition 3.1.** Signature $\Sigma_{\mathcal{T}} = \bigcup_{\alpha \in \mathcal{T}} \Sigma_\alpha$ (includes countably infinite signatures Σ_α^{Sk} of Skolem contants).
- $\mathcal{V}_{\mathcal{T}} = \bigcup_{\alpha \in \mathcal{T}} \mathcal{V}_\alpha$, such that $\mathcal{V}_\alpha$ are countably infinite.
- **Definition 3.2.** We call the set $wff_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ defined by the rules
  - $\mathcal{V}_\alpha \cup \Sigma_\alpha \subseteq wff_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$
  - If $\mathrm{C} \in wff_{\alpha \to \beta}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ and $\mathrm{A} \in wff_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$, then $\mathrm{C}\,\mathrm{A} \in wff_\beta(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$
  - If $\mathrm{A} \in wff_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$, then $\lambda X_\beta.\mathrm{A} \in wff_{\beta \to \alpha}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$

  the set of well typed formulae of type α over the signature $\Sigma_{\mathcal{T}}$ and use $wff_{\mathcal{T}}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}}) := \bigcup_{\alpha \in \mathcal{T}} wff_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ for the set of all well-typed formulae.
- **Definition 3.3.** We will call all occurrences of the variable X in A bound in $\lambda X.\mathrm{A}$. Variables that are not bound in B are called free in B.
- Substitutions are well typed, i.e. $\sigma(X_\alpha) \in wff_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ and capture-avoiding.
- **Definition 3.4 (Simply Typed λ-Calculus).** The simply typed λ calculus $\Lambda^{\to}$ over a signature $\Sigma_{\mathcal{T}}$ has the formulae $wff_{\mathcal{T}}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ (they are called λ-terms) and the following equalities:
  - α conversion: $\lambda X.\mathrm{A} =_\alpha \lambda Y.([Y/X](\mathrm{A}))$.
  - β conversion: $(\lambda X.\mathrm{A})\,\mathrm{B} =_\beta [\mathrm{B}/X](\mathrm{A})$.
  - η conversion: $\lambda X.\mathrm{A}\,X =_\eta \mathrm{A}$ if $X \notin \mathrm{free}(\mathrm{A})$.

Simply typed λ-Calculus (Notations)

- **Application is left-associative:** We abbreviate $F\ A^1\ A^2\ \ldots\ A^n$ with $F\ A^1 \ldots A^n$ eliding the brackets and further with $F\ \overline{A^n}$ in a kind of vector notation.
- **Andrews' dot Notation:** $A\ .$ stands for a left bracket whose partner is as far right as is consistent with existing brackets; i.e. $A\ .B\ C$ abbreviates $A\ (B\ C)$.
- **Abstraction is right-associative:** We abbreviate $\lambda X^1.\lambda X^2.\cdots\lambda X^n.A\cdots$ with $\lambda X^1 \ldots X^n.A$ eliding brackets, and further to $\lambda\overline{X^n}.A$ in a kind of vector notation.
- **Outer brackets:** Finally, we allow ourselves to elide outer brackets where they can be inferred.

$=_{\alpha\beta\eta}$-Equality (Overview)

- **Definition 3.5.**

  Reduction with $\left\{\begin{array}{ll} =_\beta : & (\lambda X.A)\,B \rightarrow_\beta [B/X](A) \\ =_\eta : & \lambda X.A\,X \rightarrow_\eta A \end{array}\right.$ under $=_\alpha$: $\begin{array}{c}\lambda X.A \\ =_\alpha \\ \lambda Y.([Y/X](A))\end{array}$

  The treductions can be applied at top-level (as above), but also in subterms: If $A \rightarrow_{\alpha\beta\eta} B$, then $C\,A \rightarrow_{\alpha\beta\eta} C\,B$, $A\,C \rightarrow_{\alpha\beta\eta} B\,C$, and $\lambda X.A \rightarrow_{\alpha\beta\eta} \lambda X.B$.
- **Theorem 3.6.** *β-reduction is well-typed, terminating and confluent in the presence of α-conversion.*
- **Definition 3.7 (Normal Form).** We call a λ-term A a normal form (in a reduction system $\mathcal{E}$), iff no rule (from $\mathcal{E}$) can be applied to A.
- **Corollary 3.8.** *$=_{\beta\eta}$-reduction yields unique normal forms (up to $=_\alpha$-equivalence).*

Syntactic Parts of λ-Terms

- **Definition 3.9 (Parts of λ-Terms).** We can always write a λ-term in the form $T = \lambda X^1 \ldots X^k.HA^1 \ldots A^n$, where H is not an application. We call
  - H the syntactic head of T
  - $H(A^1, \ldots, A^n)$ the matrix of T, and
  - $\lambda X^1 \ldots X^k.$ (or the sequence $X^1, \ldots, X^k$) the binder of T
- **Definition 3.10.** Head reduction always has a unique β redex

$$\lambda\overline{X^n}.(\lambda Y.A)\ B^1\ \ldots\ B^n \to_\beta^h \lambda\overline{X^n}.([B^1/Y](A))\ B^2 \ldots B^n$$

- **Theorem 3.11.** *The syntactic heads of β-normal forms are constant or variables.*
- **Definition 3.12.** Let A be a λ-term, then the syntactic head of the β-normal form of A is called the head symbol of A and written as $\mathrm{head}(A)$. We call a λ-term a j-projection, iff its head is the j^{th} bound variable.
- **Definition 3.13.** We call a λ-term a η long form, iff its matrix has base type.
- **Definition 3.14.** η Expansion makes η long forms

$$\eta[\lambda X^1 \ldots X^n.A] := \lambda X^1 \ldots X^n.\lambda Y^1 \ldots Y^m.A\ Y^1\ \ldots\ Y^m$$

- **Definition 3.15.** Long $\beta\eta$ normal form, iff it is β normal and η-long.

Semantics of $\Lambda^{\to}$

- **Definition 3.16.** We call a collection $\mathcal{D}_{\mathcal{T}} := \{\mathcal{D}_\alpha \,|\, \alpha \in \mathcal{T}\}$ a typed collection (of sets) and a collection $f_{\mathcal{T}} \colon \mathcal{D}_{\mathcal{T}} \to \mathcal{E}_{\mathcal{T}}$, a typed function, iff $f_\alpha \colon \mathcal{D}_\alpha \to \mathcal{E}_\alpha$.
- **Definition 3.17.** A typed collection $\mathcal{D}_{\mathcal{T}}$ is called a frame, iff $\mathcal{D}_{\alpha\to\beta} \subseteq \mathcal{D}_\alpha \to \mathcal{D}_\beta$.
- **Definition 3.18.** Given a frame $\mathcal{D}_{\mathcal{T}}$, and a typed function $\mathcal{I} \colon \Sigma \to \mathcal{D}$, we call $\mathcal{I}_\varphi \colon \mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}}) \to \mathcal{D}$ the value function induced by $\mathcal{I}$, iff
  1. $\mathcal{I}_\varphi|_{\mathcal{V}_{\mathcal{T}}} = \varphi$, $\quad\quad$ $\mathcal{I}_\varphi|_{\Sigma_{\mathcal{T}}} = \mathcal{I}$,
  2. $\mathcal{I}_\varphi(\mathrm{A}\,\mathrm{B}) = \mathcal{I}_\varphi(\mathrm{A})(\mathcal{I}_\varphi(\mathrm{B}))$, and
  3. $\mathcal{I}_\varphi(\lambda X_\alpha.\mathrm{A})$ is that function $f \in \mathcal{D}_{\alpha\to\beta}$, such that $f(a) = \mathcal{I}_{\varphi,[a/X]}(\mathrm{A})$ for all $a \in \mathcal{D}_\alpha$.
- **Note:** Not every λ-term has a $\mathcal{I}_\varphi$-value as we have only required $\mathcal{D}_{\alpha\to\beta} \subseteq \mathcal{D}_\alpha \to \mathcal{D}_\beta$ for frames. (there might not be enough functions)
- **Definition 3.19.** We call $\langle \mathcal{D}, \mathcal{I} \rangle$, where $\mathcal{D}$ is a frame and $\mathcal{I}$ is a typed function comprehension closed or a $\Sigma_{\mathcal{T}}$-algebra, iff $\mathcal{I}_\varphi \colon \mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}}) \to \mathcal{D}$ is total.
- **Theorem 3.20.** $=_{\alpha\beta\eta}$ *(seen as a calculus) is sound and complete for* Σ*-algebras.*
- **Upshot for LBS:** $\Lambda^{\to}$ is *the* logical system for reasoning about functions!

6.4 A Logical System for Fragment 3

Higher-Order Logic without Quantifiers ($\mathrm{HOL}^{\mathrm{nq}}$)

- **Problem:** Need a logic like $\mathrm{PL}^{\mathrm{nq}}$, but with λ-terms to interpret $\mathcal{F}_3$ into.
- **Idea:** Re-use the syntactical framework of $\Lambda^{\to}$.
- **Definition 4.1.** Let $\mathrm{HOL}^{\mathrm{nq}}$ be an instance of $\Lambda^{\to}$, with $\mathcal{BT} = \{\iota, o\}$, $\wedge \in \Sigma_{o\to o\to o}$, $\neg \in \Sigma_{o\to o}$, and $= \in \Sigma_{\alpha\to\alpha\to o}$ for all types α.
- **Idea:** To extend this to a semantics for $\mathrm{HOL}^{\mathrm{nq}}$, we only have to say something about the base type o, and the logical constants $\neg_{o\to o}$, $\wedge_{o\to o\to o}$, and $=_{\alpha\to\alpha\to o}$.
- **Definition 4.2.** We define the semantics of $\mathrm{HOL}^{\mathrm{nq}}$ by setting
  1. $\mathcal{D}_o = \{\mathsf{T}, \mathsf{F}\}$; the set of truth values
  2. $\mathcal{I}(\neg) \in \mathcal{D}_{o\to o}$, is the function $\{\mathsf{F} \mapsto \mathsf{T}, \mathsf{T} \mapsto \mathsf{F}\}$
  3. $\mathcal{I}(\wedge) \in \mathcal{D}_{o\to o\to o}$ is the function with $\mathcal{I}(\wedge)(\langle \mathsf{a}, \mathsf{b}\rangle) = \mathsf{T}$, iff $\mathsf{a} = \mathsf{T}$ and $\mathsf{b} = \mathsf{T}$.
  4. $\mathcal{I}(=) \in \mathcal{D}_{\alpha\to\alpha\to o}$ is the identity relation on $\mathcal{D}_\alpha$.

$\mathrm{HOL}^{\mathrm{nq}}$ is an expressive logical system

- $\mathrm{HOL}^{\mathrm{nq}}$ is an expressive logical system
- **Example 4.3.** We can express set union as a $\mathrm{HOL}^{\mathrm{nq}}$ term:

$$\cup := \lambda P_{\iota\to o}.\lambda Q_{\iota\to o}.\lambda X_{\iota}.P\ X \vee Q\ X$$

Let us test whether $\{1,2\} \cup \{2,3\}$ really is $\{1,2,3\}$.
Note that we can represent (the characteristic function of) $\{1,2\}$ as the $\mathrm{HOL}^{\mathrm{nq}}$ term $\lambda Z_{\iota}.Z = 1 \vee Z = 2$. (and the other sets analogously)
So lets represent $\{1,2\} \cup \{2,3\}$ as a $\mathrm{HOL}^{\mathrm{nq}}$ term and β-reduce:

$$\begin{array}{ll} & (\lambda P_{\iota\to o}.\lambda Q_{\iota\to o}.\lambda X_{\iota}.P\ X \vee Q\ X)\ (\lambda Z_{\iota}.Z = 1 \vee Z = 2)\ (\lambda Z_{\iota}.Z = 2 \vee Z = 3 \\ \to_{\beta} & (\lambda Q_{\iota\to o}.\lambda X_{\iota}.(\lambda Z_{\iota}.Z = 1 \vee Z = 2)\ X \vee Q\ X)\ (\lambda Z_{\iota}.Z = 2 \vee Z = 3) \\ \to_{\beta} & \lambda X_{\iota}.(\lambda Z_{\iota}.Z = 1 \vee Z = 2)\ X \vee (\lambda Z_{\iota}.Z = 2 \vee Z = 3)\ X \\ \to_{\beta} & \lambda X_{\iota}.X = 1 \vee X = 2 \vee X = 2 \vee X = 3 \\ \Leftrightarrow & \lambda X_{\iota}.X = 1 \vee X = 2 \vee X = 3 \end{array}$$

6.5 Translation for Fragment 3

Translations for Fragment $\mathcal{F}_3$

- We will look at the new translation rules: (the rest from $\mathcal{F}_2$ stay the same)

$$
\begin{aligned}
T1\colon & [X_{\mathrm{NP}}, Y_{VP}]_S \leadsto VP'(\mathrm{NP}'),\\
T3\colon & [X_{VP}, Y_{\mathrm{conj}}, Z_{VP}]_{VP} \leadsto \mathrm{conj}'(VP', VP'),\\
T4\colon & [X_{V^t}, Y_{\mathrm{NP}}]_{VP} \leadsto V^{t\prime}(\mathrm{NP}')
\end{aligned}
$$

- **Note:** We can get away with this because $\mathrm{PL}^{\mathrm{nq}} \subseteq \mathrm{HOL}^{\mathrm{nq}}$ in the target logic.
- The lexical insertion rules will give us two items each for *is*, *and*, and *or*, corresponding to the two types we have given them above.

word	type	term	case
BE_{pred}	$(\iota \to o) \to \iota \to o$	$\lambda P_{\iota \to o}.P$	adjective
BE_{eq}	$\iota \to \iota \to o$	$\lambda X_\iota Y_\iota.X = Y$	verb
and	$o \to o \to o$	$\wedge$	S-coord.
and	$(\iota \to o) \to (\iota \to o) \to \iota \to o$	$\lambda F_{\iota \to o} G_{\iota \to o} X_\iota.F(X) \wedge G(X)$	VP-coord.
or	$o \to o \to o$	$\vee$	S-coord.
or	$(\iota \to o) \to (\iota \to o) \to \iota \to o$	$\lambda F_{\iota \to o} G_{\iota \to o} X_\iota.F(X) \vee G(X)$	VP-coord.
didn't	$(\iota \to o) \to \iota \to o$	$\lambda P_{\iota \to o} X_\iota.\neg P\, X$	

- **Note:** All words are translated to $\mathrm{HOL}^{\mathrm{nq}}$ formulae.
- **BTW:** The translation of *or* in VP-coordination is just set union $\widehat{=}$ disjunction lifted to sets. (analogous with *and*, conjunction and intersection)

Translation Example

- It only remains to test $\mathcal{F}_3$ on an example from the original data!
- **Example 5.1.** *Ethel howled and screamed* to

$$
\begin{array}{rl}
 & (\lambda F_{\iota\to o}G_{\iota\to o}X_\iota.F(X)\wedge G(X))\ \mathrm{howls}\ \mathrm{screams}\ \mathrm{ethel} \\
\to_\beta & (\lambda G_{\iota\to o}X_\iota.\mathrm{howls}(X)\wedge G(X))\ \mathrm{screams}\ \mathrm{ethel} \\
\to_\beta & (\lambda X_\iota.\mathrm{howls}(X)\wedge \mathrm{screams}(X))\ \mathrm{ethel} \\
\to_\beta & \mathrm{howls}(\mathrm{ethel})\wedge \mathrm{screams}(\mathrm{ethel})
\end{array}
$$

6.6 Summary & Evaluation

Fragment $\mathcal{F}_3$ – Summary

- Fragment $\mathcal{F}_3$ extends $\mathcal{F}_2$ by verb phrases.
- We need a completely new idea for the logic ⇜ need functions to express translation
- Logical system: $\mathrm{HOL^{nq}} \mathrel{\widehat{=}} \Lambda^{\to} + \mathrm{PL}^0$.
  - $\Lambda^{\to}$ contributes the simple types and functions
  - PL^0 contributes type o and connectives.
- **Coverage:** Better: we can do verb phrase coordination.

Chapter 7

Fragment 4: Noun Phrases and Quantification

7.1 Fragment 4

New Data in Fragment $\mathcal{F}_4$ (more Noun Phrases)

- In $\mathcal{F}_4$ we want to extend $\mathcal{F}_3$ so it can deal with the following sentences: (without the "the-NP" trick)
  1. *Peter loved the cat.*, but not * *Peter loved the the cat.*
  2. *John killed a cat with a white tail.*
  3. *Peter chased the gangster in the red sportscar.*
  4. *Peter loves every cat.*
  5. *Every man loves a woman.*
  6. *The quick brown fox jumps over the lazy dog.*
  7. *The very heavy boat sank quickly.*

New Grammar in Fragment $\mathcal{F}_4$ (Common Noun Phrases)

- To account for the syntax we extend the functionality of noun phrases from $\mathcal{F}_1$.
- **Definition 1.1.** $\mathcal{F}_4$ adds the rules on the right to $\mathcal{F}_3$ (on the left):

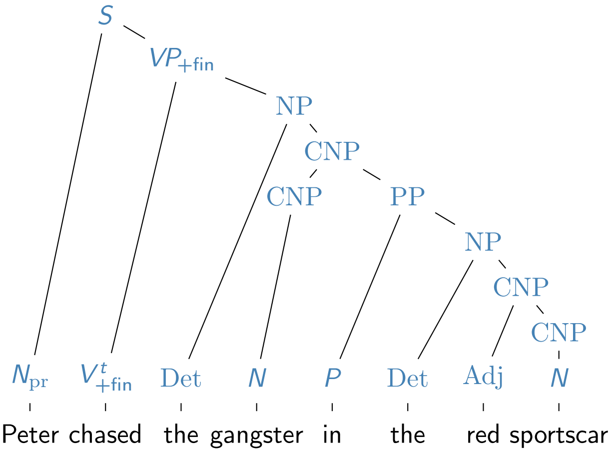
$S1$: $S \to \mathrm{NP}\ VP_{+\mathrm{fin}}$,
$S2$: $S \to S\ \mathrm{conj}\ S$,
$V1$: $VP_{\pm\mathrm{fin}} \to V^{i}_{\pm\mathrm{fin}}$,
$V2$: $VP_{\pm\mathrm{fin}} \to V^{t}_{\pm\mathrm{fin}}\ \mathrm{NP}$,
$V4$: $VP_{+\mathrm{fin}} \to BE_{=}\ \mathrm{NP}$,
$V5$: $VP_{+\mathrm{fin}} \to BE_{pred}\ \mathrm{Adj}$,
$V6$: $VP_{+\mathrm{fin}} \to$ didn't $VP_{\text{-fin}}$,
$N1$: $\mathrm{NP} \to N_{\mathrm{pr}}$,
$N2$: $\mathrm{NP} \to \mathrm{Pron}$

$S3$: $S \to S\ \mathrm{PP}$,
$N3$: $\mathrm{NP} \to \mathrm{Det}\ \mathrm{CNP}$,
$N4$: $\mathrm{CNP} \to N$,
$N5$: $\mathrm{CNP} \to \mathrm{CNP}\ \mathrm{PP}$,
$N6$: $\mathrm{CNP} \to \mathrm{Adj}\ \mathrm{CNP}$,
$P1$: $\mathrm{PP} \to P\ \mathrm{NP}$,
$V3'$: $VP_{\pm\mathrm{fin}} \to VP_{\pm\mathrm{fin}}\ \mathrm{VPconj}_{\pm\mathrm{fin}}$,
$V7$: $\mathrm{VPconj}_{\pm\mathrm{fin}} \to \mathrm{conj}\ VP_{\pm\mathrm{fin}}$,
$V8$: $VP_{+\mathrm{fin}} \to VP_{+\mathrm{fin}}\ \mathrm{Adv}$,
$V9$: $VP_{\pm\mathrm{fin}} \to VP_{\pm\mathrm{fin}}\ \mathrm{PP}$,
$L1$: $P \to \mathrm{with} \mid \mathrm{of} \mid \ldots$

- **Definition 1.2.** A common noun is a noun that describes a type, for example *woman*, or *philosophy* rather than an token, such as *Amelia Earhart* (proper name).

Testing the $\mathcal{F}_4$ Syntax on an example

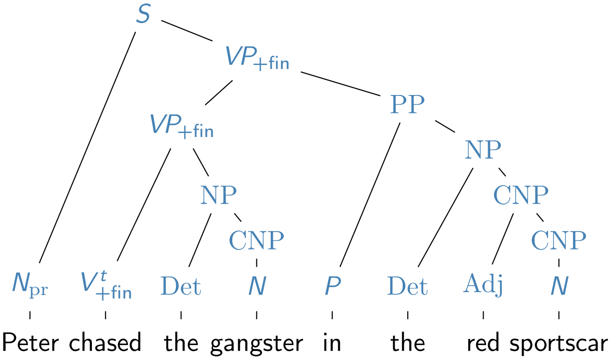
- **Example 1.3.** Can we capture the (syntactic) attachment ambiguity in *Peter chased the gangster in the red sportscar.*



$\rightsquigarrow$ The gangster is in the car

Testing the $\mathcal{F}_4$ Syntax on an example

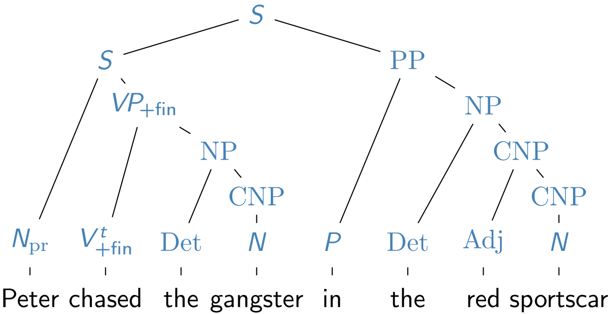
- **Example 1.4.** Can we capture the (syntactic) attachment ambiguity in *Peter chased the gangster in the red sportscar.*



$\leadsto$ Peter is in the car

Testing the $\mathcal{F}_4$ Syntax on an example

- **Example 1.5.** Can we capture the (syntactic) attachment ambiguity in *Peter chased the gangster in the red sportscar.*



⇝ Both Peter and the gangster are in the car

7.2 A Target Logic for Fragment 4

Higher-Order Logic with Descriptions

- **Plan:** We need to extend $\mathrm{HOL}^{\mathrm{nq}}$ with
  - quantifiers so we can treat *Every student sleeps*
  - a logical operator for definite descriptions, e.g. *the teacher sleeps*

  We will call this logic Higher-Order Logic with Descriptions (quantifiers taken for granted)

Higher-Order Logic with Descriptions

- **Plan:** We need to extend $\mathrm{HOL}^{\mathrm{nq}}$ with
  - quantifiers so we can treat *Every student sleeps*
  - a logical operator for definite descriptions, e.g. *the teacher sleeps*

  We will call this logic Higher-Order Logic with Descriptions (quantifiers taken for granted)
- **Note:** Quantifiers can be added to any logic: Extend the
  - syntax by variables and a new binding symbol (language-level)
  - semantics by a new clause for the value function
  - calculi by new quantifier introduction/elimination rules

  Quite tedious compared to simply adding a new logical constant!

Higher-Order Logic with Descriptions

- **Plan:** We need to extend $\mathrm{HOL}^{\mathrm{nq}}$ with
  - quantifiers so we can treat *Every student sleeps*
  - a logical operator for definite descriptions, e.g. *the teacher sleeps*

  We will call this logic Higher-Order Logic with Descriptions (quantifiers taken for granted)
- **Note:** Quantifiers can be added to any logic: Extend the
  - syntax by variables and a new binding symbol (language-level)
  - semantics by a new clause for the value function
  - calculi by new quantifier introduction/elimination rules

  Quite tedious compared to simply adding a new logical constant!
- **Note:** The description operator will have to have type $(\iota \to o) \to \iota$, as the denotation of *teacher* has type $\iota \to o$ and *the teacher* has type ι. (like *Mary*)

7.2.1 Quantifiers and Equality in Higher-Order Logic

Higher-Order Abstract Syntax

- **Idea:** In $\mathrm{HOL}^{\to}$, we already have binding operator: λ, use that to treat quantification.
- **Definition 2.1.** We add two new logical constants Π^α and Σ^α for each type α:
  1. $\mathcal{I}(\Pi^\alpha)(p) = \mathsf{T}$, iff $p(a) = \mathsf{T}$ for all $a \in \mathcal{D}_\alpha$ (i.e. if p is the universal set)
  2. $\mathcal{I}(\Sigma^\alpha)(p) = \mathsf{T}$, iff $p(a) = \mathsf{T}$ for some $a \in \mathcal{D}_\alpha$ (i.e. iff p is non-empty)
- **Definition 2.2.** Regain traditional quantifiers as abbreviations:

$$(\forall X_\alpha.A) := \Pi^\alpha\,(\lambda X_\alpha.A) \qquad (\exists X_\alpha.A) := \Sigma^\alpha\,(\lambda X_\alpha.A)$$

- **Observation:** Indeed: $\mathcal{I}_\varphi(\forall X_\iota.A) = \mathcal{I}_\varphi(\Pi^\iota\,(\lambda X_\iota.A)) = \mathcal{I}(\Pi^\iota)(\mathcal{I}_\varphi(\lambda X_\iota.A)) = \mathsf{T}$ iff $\mathcal{I}_\varphi(\lambda X_\iota.A)(a) = \mathcal{I}_{[a/X],\varphi}(A) = \mathsf{T}$ for all $a \in \mathcal{D}_\alpha$.
- **Definition 2.3.** We call this approach to binding operators higher-order abstract syntax (HOAS).

Equality

- **Definition 2.4 (Leibniz equality).** $Q^\alpha A_\alpha B_\alpha = \forall P_{\alpha\to o}.PA \Leftrightarrow PB$ (Leibniz' indiscernibility of identicals)
- **Note:** $\forall P_{\alpha\to o}.PA \Rightarrow PB$ (get the other direction by instantiating P with Q, where $QX \Leftrightarrow \neg PX$)
- **Theorem 2.5.** *If $\mathcal{M} = \langle\mathcal{D},\mathcal{I}\rangle$ is a standard model, then $\mathcal{I}_\varphi(Q^\alpha)$ is the identity relation on $\mathcal{D}_\alpha$.*
- **Definition 2.6 (Notation).** We write $A = B$ for QAB (A and B are equal, iff there is no property P that can tell them apart.)
- *Proof:*
  1. $\mathcal{I}_\varphi(QAB) = \mathcal{I}_\varphi(\forall P.PA \Rightarrow PB) = T$, iff $\mathcal{I}_{\varphi,[r/P]}(PA \Rightarrow PB) = T$ for all $r \in \mathcal{D}_{\alpha\to o}$.
  2. For $A = B$ we have $\mathcal{I}_{\varphi,[r/P]}(PA) = r(\mathcal{I}_\varphi(A)) = F$ or $\mathcal{I}_{\varphi,[r/P]}(PB) = r(\mathcal{I}_\varphi(B)) = T$.
  3. Thus $\mathcal{I}_\varphi(QAB) = T$.
  4. Let $\mathcal{I}_\varphi(A) \neq \mathcal{I}_\varphi(B)$ and $r = \{\mathcal{I}_\varphi(A)\} \in \mathcal{D}_{\alpha\to o}$ (exists in a standard model)
  5. so $r(\mathcal{I}_\varphi(A)) = T$ and $r(\mathcal{I}_\varphi(B)) = F$
  6. $\mathcal{I}_\varphi(QAB) = F$, as $\mathcal{I}_{\varphi,[r/P]}(PA \Rightarrow PB) = F$, since $\mathcal{I}_{\varphi,[r/P]}(PA) = r(\mathcal{I}_\varphi(A)) = T$ and $\mathcal{I}_{\varphi,[r/P]}(PB) = r(\mathcal{I}_\varphi(B)) = F$.

Alternative: HOL^∞

- **Definition 2.7.** There is only one logical constant in HOL^∞: $q^\alpha \in \Sigma_{\alpha\to\alpha\to o}$ with $\mathcal{I}(q^\alpha)(a,b) = \mathsf{T}$, iff $a = b$.
  We define the rest as below: Definitions (D) and Notations (N)

N	$\mathrm{A}_\alpha = \mathrm{B}_\alpha$	for	$q^\alpha \mathrm{A}_\alpha \mathrm{B}_\alpha$
D	T	for	$q^o = q^o$
D	F	for	$\lambda X_o.T = \lambda X_o.X_o$
D	Π^α	for	$q^{\alpha\to o}\ (\lambda X_\alpha.T)$
N	$\forall X_\alpha.\mathrm{A}$	for	$\Pi^\alpha\ (\lambda X_\alpha.\mathrm{A})$
D	$\wedge$	for	$\lambda X_o.\lambda Y_o.(\lambda G_{o\to o\to o}.G T T = \lambda G_{o\to o\to o}.G X Y)$
N	$\mathrm{A} \wedge \mathrm{B}$	for	$\wedge\ (\mathrm{A}_o)\ (\mathrm{B}_o)$
D	$\Rightarrow$	for	$\lambda X_o.\lambda Y_o.(X = X \wedge Y)$
N	$\mathrm{A} \Rightarrow \mathrm{B}$	for	$\Rightarrow (\mathrm{A}_o)\ (\mathrm{B}_o)$
D	$\neg$	for	$q^o\ F$
D	$\vee$	for	$\lambda X_o.\lambda Y_o.\neg(\neg X \wedge \neg Y)$
N	$\mathrm{A} \vee \mathrm{B}$	for	$\vee\ (\mathrm{A}_o)\ (\mathrm{B}_o)$
D	$\exists X_\alpha.\mathrm{A}_o$	for	$\neg(\forall X_\alpha.\neg\mathrm{A})$
N	$\mathrm{A}_\alpha \neq \mathrm{B}_\alpha$	for	$\neg q^\alpha\ (\mathrm{A}_\alpha)\ (\mathrm{B}_\alpha)$

- yield the intuitive meanings for connectives and quantifiers.

7.2.2 A Logic for Definite Descriptions

Semantics of Definite Descriptions

- **Problem:** We need the meaning for the determiner *the*, as in *the boy runs*
- **Idea (Type):** *the boy* behaves like a proper name (e.g. *Peter*), i.e. has type ι. Applying *the* to a noun (type $\iota \to o$) yields ι. So *the* has type $(\alpha \to o) \to \alpha$, i.e. it takes a set as argument.
- **Idea (Semantics):** *the* has the fixed semantics that this function returns the single member of its argument if the argument is a singleton, and is otherwise undefined. (new logical constant)
- **Definition 2.8.** We introduce a new logical constant ι. $\mathcal{I}(\iota)$ is the function $f \in \mathcal{D}_{(\alpha \to o) \to \alpha}$, such that $f(s) = a$, iff $s \in \mathcal{D}_{\alpha \to o}$ is the singleton $\{a\}$, and is otherwise undefined. (remember that we can interpret predicates as sets)
- **Axioms for ι:**

$$\forall X_\alpha . X = \iota \;\; = \;X$$
$$\forall P, Q . Q(\iota \;\; P) \wedge (\forall X, Y . P(X) \wedge P(Y) \Rightarrow X = Y) \Rightarrow (\forall . P(Z) \Rightarrow Q(Z))$$

More Operators and Axioms for $\mathrm{HOL}^{\rightarrow}$

- **Definition 2.9.** The unary conditional $\mathrm{w}^{\alpha} \in \Sigma_{o\rightarrow\alpha\rightarrow\alpha}$
  $\mathrm{w}\ (\mathrm{A}_{o})\mathrm{B}_{\alpha}$ means: “If A, then B”.
- **Definition 2.10.** The binary conditional $\mathrm{if}^{\alpha} \in \Sigma_{o\rightarrow\alpha\rightarrow\alpha\rightarrow\alpha}$
  $\mathrm{if}\ (\mathrm{A}_{o})\ (\mathrm{B}_{\alpha})\ (\mathrm{C}_{\alpha})$ means: “if A, then B else C”.
- **Definition 2.11.** The description operator $\iota^{\alpha} \in \Sigma_{(\alpha\rightarrow o)\rightarrow\alpha}$
  if P is a singleton set, then $\iota\ (\mathrm{P}_{\alpha\rightarrow o})$ is the (unique) element in P.
- **Definition 2.12.** The choice operator $\gamma^{\alpha} \in \Sigma_{(\alpha\rightarrow o)\rightarrow\alpha}$
  if P is non-empty, then $\gamma\ (\mathrm{P}_{\alpha\rightarrow o})$ is an arbitrary element from P.
- **Definition 2.13 (Axioms for these Operators).**
  - unary conditional: $\forall\varphi_{o}.\forall X_{\alpha}.\varphi \Rightarrow \mathrm{w}\ \varphi X = X$
  - binary conditional: $\forall\varphi_{o}.\forall X_{\alpha}, Y_{\alpha}, Z_{\alpha}.(\varphi \Rightarrow \mathrm{if}\ \varphi\ X\ Y = X) \wedge (\neg\varphi \Rightarrow \mathrm{if}\ \varphi\ Z\ X = X)$
  - description operator $\forall P_{\alpha\rightarrow o}.(\exists^{1} X_{\alpha}.PX) \Rightarrow (\forall Y_{\alpha}.PY \Rightarrow \iota\ P = Y)$
  - choice operator $\forall P_{\alpha\rightarrow o}.(\exists X_{\alpha}.PX) \Rightarrow (\forall Y_{\alpha}.PY \Rightarrow \gamma\ P = Y)$
- **Idea:** These operators ensure a much larger supply of functions in Henkin models.

More on the Description Operator

- ι is a weak form of the choice operator. (only works on singletons)
- Alternative Axiom of Descriptions: $\forall X_\alpha.\iota^\alpha \;\; = X = X$.
  - use that $\mathcal{I}_{[\mathsf{a}/X]}(= \; X) = \{\mathsf{a}\}$
  - we only need this for base types $\neq o$
  - Define $\iota^o \;\; := = \;\; (\lambda X_o.X)$ or $\iota^o \;\; := \lambda G_{o\to o}.G \; T$ or $\iota^o \;\; := = = \;\; T$
  - $\iota^{(\alpha\to\beta)} \;\; := \lambda H_{(\alpha\to\beta)\to o} X_\alpha.\iota^\beta \;\; (\lambda Z_\beta.(\exists F_{\alpha\to\beta}.H \; F \wedge F \; X = Z))$

7.3 Translation for Fragment 4

Translation of Determiners and Quantifiers

- **Idea:** We establish the meaning of quantifying determiners by $=_\beta$-expansion.
  1. assume that we are translating into a λ-calculus with quantifiers and that
     - $\forall X.\mathrm{boy}(X) \Rightarrow \mathrm{runs}(X)$ translates *Every boy runs*, and
     - $\exists X.\mathrm{boy}(X) \wedge \mathrm{runs}(X)$ for *Some boy runs*
  2. $\forall\!\!\!\forall := \lambda P_{\iota\to o} Q_{\iota\to o}.(\forall.P(X) \Rightarrow Q(X))$ for *every*. (subset relation)
  3. $\exists\!\!\!\exists := \lambda P_{\iota\to o} Q_{\iota\to o}.(\exists.P(X) \wedge Q(X))$ for *some*. (non-empty intersection)
- **Problem:** Linguistic quantifiers take two arguments (restriction and scope), logical ones only one! (in logics, restriction is the universal set)
- We cannot treat *the* with regular quantifiers (new logical constant; see below)
- **Definition 3.1.**
  We translate the word the to $\tau := \lambda P_{\iota\to o} Q_{\iota\to o}.Q\ \iota\ P$, where ι is a new operator that given a set returns its (unique) member.
- **Example 3.2.** This translates *The pope spoke* to $\tau(\mathrm{pope}, \mathrm{speaks})$, which $=_\beta$-reduces to $\mathrm{speaks}(\iota\ \mathrm{pope})$.

Translation of Special lexical items and classes

- If *Adj* is an intersective adjective and Adj' is an constant of type $\iota \rightarrow o$, then
  - 9: Adj $\rightsquigarrow Adj'$ or
  - 9′: Adj $\rightsquigarrow (\lambda P_{\iota \rightarrow o} X_{\iota}.P(X) \wedge Adj'(X))$
- If *Adj* is a non-intersective adjective, then Adj' is a constant of type $(\iota \rightarrow o) \rightarrow \iota \rightarrow o$ whose denotation is given the interpretation by $\mathcal{I}$ and
  - 10: Adj $\rightsquigarrow Adj'$.

Proper names

- **Problem:** Subject NPs with quantificational determiners have type $(\iota \to o) \to o$ (and are applied to the VP) whereas subject NPs with proper names have type ι. (argument to the VP)
- **Idea:** *John runs* translates to $\mathrm{runs}(\mathrm{john})$, where $\mathrm{runs} \in \Sigma_{\iota \to o}$ and $\mathrm{john} \in \Sigma_{\iota}$.
  Now we $=_\beta$-expand over the VP yielding $(\lambda P_{\iota \to o}.P(\mathrm{john}))\ \mathrm{runs}$
  $\lambda P_{\iota \to o}.P(\mathrm{john})$ has type $(\iota \to o) \to o$ and can be applied to the VP runs.
- **Definition 3.3.** If $c \in \Sigma_\alpha$, then type raising c yields $\lambda P_{\alpha \to o}.P\ c$.

Definite NPs

- **Problem:** On our current assumptions, $\mathit{the}' = \iota$, and so for any definite NP *the N*, its translation is $\iota\ N$, an expression of type ι.
- **Idea:** Type lift just as we did with proper names: $\iota\ N$ type lifts to $\lambda P.P\ \iota\ N$, so $\mathit{the}' = \lambda PQ.Q\ \iota\ P$
- **Advantage:** This is a "generalized quantifier treatment": the' treated as denoting relations between sets.
- **Solution by Barwise&Cooper 1981:** For any $\mathrm{a} \in \mathcal{D}_{\iota\to o}$:
  $\mathcal{I}(\mathit{the}')(\mathrm{a}) = \mathcal{I}(\mathit{every}')(\mathrm{a})$ if $\#(\mathrm{a}) = 1$, undefined otherwise
  So the' is that function in $\mathcal{D}_{(\iota\to o)\to(\iota\to o)\to o}$ such that for any $A, B \in \mathcal{D}_{\iota\to o}$
  if $\#(A) = 1$ then $\mathit{the}'(A, B) = \mathsf{T}$ if $A \subseteq B$ and $\mathit{the}'(A, B) = \mathsf{F}$ if $A \not\subseteq B$ otherwise undefined

Problems with Type raised NPs

- **Problem:** We have type-raised NPs, but consider transitive verbs as in *Mary loves most cats*. loves is of type $\iota \to \iota \to o$ while the object NP is of type $(\iota \to o) \to o$ (application?)
- **Another Problem:** We encounter the same problem in the sentence *Mary loves John* if we choose to type-lift the NPs.
- **Idea:** Change the type of the transitive verb to allow it to "swallow" the higher-typed object NP.
- **Better Idea:** Adopt a new rule for semantic composition for this case.
- **Remember:** $loves'$ is a function from individuals (e.g. *John*) to properties (in the case of the VP *loves John*, the property X *loves John* of X).

Type raised NPs and Function Composition

- We can extend $\mathrm{HOL}^{\rightarrow}$ by a constant $\circ_{(\beta\rightarrow\gamma)\rightarrow(\alpha\rightarrow\beta)\rightarrow\alpha\rightarrow\gamma}$ by setting $\circ := \lambda FGX.F(G(X))$ thus

$$\circ\; g\; f \rightarrow_{\beta} \lambda X.g(f(X)) \qquad \text{and} \qquad \circ\; g\; f\; a \rightarrow_{\beta} g(f(a))$$

In our example, we have

$$\begin{aligned} \circ\;(\lambda P.P(\mathrm{john}))\;\mathrm{loves} \quad &=_{Def} \quad (\lambda FGX.F(G(X)))\;(\lambda P.P(\mathrm{john}))\;\mathrm{loves} \\ &\rightarrow_{\beta} \quad (\lambda GX.(\lambda P.P(\mathrm{john}))\;G(X))\;\mathrm{loves} \\ &\rightarrow_{\beta} \quad \lambda X.(\lambda P.P(\mathrm{john}))\;\mathrm{loves}\;X \\ &\rightarrow_{\beta}! \quad \lambda X.\mathrm{loves}(X,\mathrm{john}) \end{aligned}$$

Generalized Quantifiers

- **Problem:** What about *Most boys run.*: linguistically *most* behaves exactly like *every* or *some*.
- **Idea:** *Most boys run* is true just in case the number of boys who run is greater than the number of boys who do not run.

$$\#(\mathcal{I}_\varphi(\mathrm{boy}) \cap \mathcal{I}_\varphi(\mathrm{runs})) > \#(\mathcal{I}_\varphi(\mathrm{boy}) \backslash \mathcal{I}_\varphi(\mathrm{runs}))$$

- **Definition 3.4.** $\#(A) > \#(B)$, iff there is no surjective function from B to A, so we can define

$$most' := \lambda AB.\neg(\exists F.\forall X.A(X) \wedge \neg B(X) \Rightarrow (\exists.A(Y) \wedge B(Y) \wedge X = F(Y)))$$

Back to *every* and *some* (set characterization)

- We can now give an explicit set characterization of *every* and *some*:
  1. *every* denotes $\{\langle X, Y\rangle \mid X \subseteq Y\}$
  2. *some* denotes $\{\langle X, Y\rangle \mid X \cap Y \neq \emptyset\}$
- The denotations can be given in equivalent function terms, as demonstrated above with the denotation of *most*.

7.4 Inference for Fragment 4

7.4.1 Model Generation with Quantifiers

Model Generation (The *RM* Calculus [Kon04])

- **Idea:** Try to generate domain-minimal (i.e. fewest individuals) Herbrand models (for NL interpretation)
- **Problem:** Even one function constant makes Herbrand universe infinite (solution: leave them out)
- **Definition 4.1.** *RM* adds ground quantifier rules to propositional tableau calculus

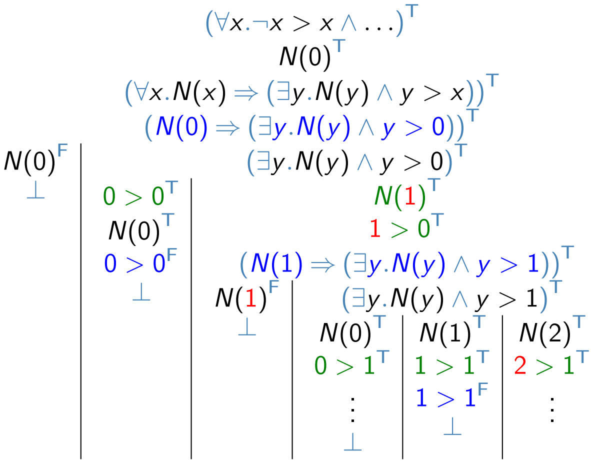
$$\frac{(\forall X.\mathrm{A})^{\mathsf{T}} \quad c \in \mathcal{H}}{([c/X](\mathrm{A}))^{\mathsf{T}}} \; RM{\forall}$$

$$\frac{(\forall X.\mathrm{A})^{\mathsf{F}} \quad \mathcal{H} = \{a_1, \ldots, a_n\} \quad w \notin \mathcal{H} \text{ new}}{([a_1/X](\mathrm{A}))^{\mathsf{F}} \;\Big|\; \ldots \;\Big|\; ([a_n/X](\mathrm{A}))^{\mathsf{F}} \;\Big|\; ([w/X](\mathrm{A}))^{\mathsf{F}}} \; RM{\exists}$$

  - *RM*∃ rule introduces new witness constant w to the branch Herbrand universe $\mathcal{H}$: the set of all individual constants on the branch.
  - Apply *RM*∀ exhaustively (for new w reapply all *RM*∀ rules on branch!)

Generating infinite models (Natural Numbers)

- We have to re-apply the $RM\forall$ rule for any new constant
- **Example 4.2.** This leads to the generation of infinite models



$$\begin{array}{c}
(\forall x.\neg x > x \land \ldots)^{T} \\
N(0)^{T} \\
(\forall x.N(x) \Rightarrow (\exists y.N(y) \land y > x))^{T} \\
(N(0) \Rightarrow (\exists y.N(y) \land y > 0))^{T} \\
\begin{array}{c|c}
\begin{array}{c} N(0)^{F} \\ \bot \end{array} &
\begin{array}{c}
(\exists y.N(y) \land y > 0)^{T} \\
\begin{array}{c|c}
\begin{array}{c} 0 > 0^{T} \\ N(0)^{T} \\ 0 > 0^{F} \\ \bot \end{array} &
\begin{array}{c}
N(1)^{T} \\
1 > 0^{T} \\
(N(1) \Rightarrow (\exists y.N(y) \land y > 1))^{T} \\
\begin{array}{c|c}
\begin{array}{c} N(1)^{F} \\ \bot \end{array} &
\begin{array}{c}
(\exists y.N(y) \land y > 1)^{T} \\
\begin{array}{c|c|c}
\begin{array}{c} N(0)^{T} \\ 0 > 1^{T} \\ \vdots \\ \bot \end{array} &
\begin{array}{c} N(1)^{T} \\ 1 > 1^{T} \\ 1 > 1^{F} \\ \bot \end{array} &
\begin{array}{c} N(2)^{T} \\ 2 > 1^{T} \\ \vdots \end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}$$

Example: *Peter is a man. No man walks*

- **Example 4.3 (Model generation with quantifiers).**
  Peter is a man. No man walks

$$
\begin{array}{c}
\boxed{\mathrm{man}(\mathrm{peter})}\\
\boxed{\neg(\exists X.\mathrm{man}(X)\wedge\mathrm{walks}(X))}\\
(\exists X.\mathrm{man}(X)\wedge\mathrm{walks}(X))^{\mathsf{F}}\\
(\forall X.\neg\mathrm{man}(X)\vee\neg\mathrm{walks}(X))^{\mathsf{T}}\\
(\neg\mathrm{man}(\mathrm{peter})\vee\neg\mathrm{walks}(\mathrm{peter}))^{\mathsf{T}}\\
\begin{array}{c|c}
\neg\mathrm{man}(\mathrm{peter})^{\mathsf{T}} & \neg\mathrm{walks}(\mathrm{peter})^{\mathsf{T}}\\
\mathrm{man}(\mathrm{peter})^{\mathsf{F}} & \mathrm{walks}(\mathrm{peter})^{\mathsf{F}}\\
\bot &
\end{array}\\
\text{Herbrand valuation: } \{\mathrm{man}(\mathrm{peter})^{\mathsf{T}},\mathrm{walks}(\mathrm{peter})^{\mathsf{F}}\}
\end{array}
$$

Anaphor Resolution *A man sleeps. He snores*

- **Example 4.4 (Anaphor Resolution).** *A man sleeps. He snores*

$$\begin{array}{c}
\boxed{\exists X.\mathrm{man}(X) \land \mathrm{sleeps}(X)} \\
(\mathrm{man}(c^1) \land \mathrm{sleeps}(c^1))^{\mathsf{T}} \\
\mathrm{man}(c^1)^{\mathsf{T}} \\
\mathrm{sleeps}(c^1)^{\mathsf{T}} \\
\boxed{\exists Y.\mathrm{man}(Y) \land \mathrm{snores}(Y)} \\
\begin{array}{c|c}
(\mathrm{man}(c^1) \land \mathrm{snores}(c^1))^{\mathsf{T}} & (\mathrm{man}(c^2) \land \mathrm{snores}(c^2))^{\mathsf{T}} \\
\mathrm{man}(c^1)^{\mathsf{T}} & \mathrm{man}(c^2)^{\mathsf{T}} \\
\mathrm{snores}(c^1)^{\mathsf{T}} & \mathrm{snores}(c^2)^{\mathsf{T}} \\
\textit{minimal} & \textit{deicbtic}
\end{array}
\end{array}$$

Anaphora with World Knowledge

- **Example 4.5.** *Mary is married to Jeff. Her husband is not in town.* (slightly outside $\mathcal{F}_2$)
  In PL^1: $\mathrm{married}(\mathrm{mary},\mathrm{jeff})$, and

$$\exists W_{\mathbb{M}\mathtt{ale}}, W'_{\mathbb{F}\mathtt{emale}}.\mathrm{husband}(W,W') \wedge \neg\mathrm{intown}(W)$$

- World knowledge
  - If woman X is married to man Y, then Y is the only husband of X.
  - $\forall X_{\mathbb{F}\mathtt{emale}}, Y_{\mathbb{M}\mathtt{ale}}.\mathrm{married}(X,Y) \Rightarrow \mathrm{husband}(Y,X) \wedge (\forall Z.\mathrm{husband}(Z,X) \Rightarrow (Z = Y))$
- Model generation gives tableau where all open branches contain

$$\{\mathrm{married}(\mathrm{mary},\mathrm{jeff})^{\mathsf{T}}, \mathrm{husband}(\mathrm{jeff},\mathrm{mary})^{\mathsf{T}}, \mathrm{intown}(\mathrm{jeff})^{\mathsf{F}}\}$$

- **Differences:** Additional negative facts e.g. $\mathrm{married}(\mathrm{mary},\mathrm{mary})^{\mathsf{F}}$.

A branch without World Knowledge

$$\mathrm{married}(\mathrm{mary},\mathrm{jeff})^{\mathsf{T}}$$
$$(\exists Z_{\mathbb{M}\mathtt{ale}}, Z'_{\mathbb{F}\mathtt{emale}}.\mathrm{husband}(Z,Z') \wedge \neg\mathrm{intown}(Z))^{\mathsf{T}}$$
$$(\exists Z'.\mathrm{husband}({c^1}_{\mathbb{M}\mathtt{ale}}, Z') \wedge \neg\mathrm{intown}({c^1}_{\mathbb{M}\mathtt{ale}}))^{\mathsf{T}}$$
$$(\mathrm{husband}({c^1}_{\mathbb{M}\mathtt{ale}}, \mathrm{mary}) \wedge \neg\mathrm{intown}({c^1}_{\mathbb{M}\mathtt{ale}}))^{\mathsf{T}}$$
$$\mathrm{husband}({c^1}_{\mathbb{M}\mathtt{ale}}, \mathrm{mary})^{\mathsf{T}}$$
$$\neg\mathrm{intown}({c^1}_{\mathbb{M}\mathtt{ale}})^{\mathsf{T}}$$
$$\mathrm{intown}({c^1}_{\mathbb{M}\mathtt{ale}})^{\mathsf{F}}$$

- **Problem:** Bigamy: ${c^1}_{\mathbb{M}\mathtt{ale}}$ and jeff are husbands of *Mary*!

7.4.2 Model Generation with Definite Descriptions

A Model Generation Rule for ι

- **Definition 4.6.**

$$\frac{\begin{array}{c}P(c)^{\mathsf{T}}\\ Q(\iota\ P)^{\alpha}\end{array}\quad \mathcal{H}=\{c,a_1,\ldots,a_n\}}{\begin{array}{c}Q(c)^{\alpha}\\ (P(a_1)\Rightarrow c=a_1)^{\mathsf{T}}\\ \vdots\\ (P(a_n)\Rightarrow c=a_n)^{\mathsf{T}}\end{array}}\ RM\,\iota$$

- **Intuition:** If we have a member c of P and $Q(\iota\ P)$ is defined (it has truth value $\alpha\in\{\mathsf{T},\mathsf{F}\}$), then P must be a singleton (i.e. all other members X of P are identical to c) and Q must hold on c. So the rule $RM\,\iota$ forces it to be by making all other members of P equal to c.

Mary owned a lousy computer. The hard drive crashed.

$$\begin{array}{c}
(\forall X.\mathrm{computer}(X) \Rightarrow (\exists Y.\mathrm{harddrive}(Y) \wedge \mathrm{partof}(Y,X)))^{\mathsf{T}} \\
\boxed{\exists X.\mathrm{computer}(X) \wedge \mathrm{lousy}(X) \wedge \mathrm{own}(\mathrm{mary},X)} \\
\mathrm{computer}(c)^{\mathsf{T}} \\
\mathrm{lousy}(c)^{\mathsf{T}} \\
\mathrm{own}(\mathrm{mary},c)^{\mathsf{T}} \\
\begin{array}{c|c}
\mathrm{harddrive}(c)^{\mathsf{T}} & \mathrm{harddrive}(d)^{\mathsf{T}} \\
\mathrm{partof}(c,c)^{\mathsf{T}} & \mathrm{partof}(d,c)^{\mathsf{T}} \\
\vdots & \boxed{\mathrm{crashes}(\iota\ \mathrm{harddrive})} \\
\bot & \mathrm{crashes}(d)^{\mathsf{T}} \\
 & (\mathrm{harddrive}(\mathrm{mary}) \Rightarrow \mathrm{mary} = d)^{\mathsf{T}} \\
 & (\mathrm{harddrive}(c) \Rightarrow c = d)^{\mathsf{T}}
\end{array}
\end{array}$$

Another Example *The dog barks*

- In a situation, where there are two dogs: Fido and Chester

$$\begin{array}{c}
\mathrm{dog}(\mathrm{fido})^{\mathsf{T}}\\
\mathrm{dog}(\mathrm{chester})^{\mathsf{T}}\\
\boxed{\mathrm{bark}(\iota\ \mathrm{dog})}\\
\mathrm{bark}(\mathrm{fido})^{\mathsf{T}}\\
(\mathrm{dog}(\mathrm{chester}) \Rightarrow \mathrm{chester} = \mathrm{fido})^{\mathsf{T}}\\
\begin{array}{c|c}
\mathrm{dog}(\mathrm{chester})^{\mathsf{F}} & \mathrm{chester} = \mathrm{fido}^{\mathsf{T}}\\
\bot &
\end{array}
\end{array} \tag{1}$$

- Note that none of our rules allows us to close the right branch, since we do not know that Fido and Chester are distinct. Indeed, they could be the same dog (with two different names). But we can eliminate this possibility by adopting a new assumption.

7.4.3 Model Generation with Unique Name Assumptions

Model Generation with Unique Name Assumption (UNA)

- **Problem:** Names are unique usually in natural language
- **Definition 4.7.** The unique name assumption (UNA) makes the assumption that names are unique (in the respective context)
- **Idea:** Add background knowledge of the form $n = m^{\mathsf{F}}$ (*n* and *m* names)
- **Better Idea:** Build UNA into the calculus: partition the Herbrand universe $\mathcal{H} = \mathcal{U} \cup \mathcal{W}$ into subsets $\mathcal{U}$ for constants with a UNA, and $\mathcal{W}$ without. (treat them differently)
- **Definition 4.8 (Model Generation with UNA).** We add the following two rules to the *RM* calculus to deal with the unique name assumption.

$$\frac{\begin{array}{c} a = b^{\mathsf{T}} \\ \mathsf{A}^{\alpha} \end{array} \quad a \in \mathcal{W} \quad b \in \mathcal{H}}{([b/a](\mathsf{A}))^{\alpha}} \; \mathit{RM}\,\mathrm{subst} \qquad \frac{a = b^{\mathsf{T}} \quad a, b \in \mathcal{U}}{\bot} \; \mathit{RM}\,\mathrm{una}$$

Solving a Crime with Unique Names

- **Example 4.9.** Tony has observed (at most) two people. Tony observed a murderer that had black hair. It turns out that Bill and Bob were the two people Tony observed. Bill is blond, and Bob has black hair. (Who was the murderer.) Let $\mathcal{U} = \{\mathrm{Bill}, \mathrm{Bob}\}$ and $\mathcal{W} = \{\mathrm{murderer}\}$:

$$\begin{array}{c}
(\forall z.\mathrm{observes}(\mathrm{Tony}, z) \Rightarrow (z = \mathrm{Bill} \lor z = \mathrm{Bob}))^{\mathsf{T}} \\
\mathrm{observes}(\mathrm{Tony}, \mathrm{Bill})^{\mathsf{T}} \\
\mathrm{observes}(\mathrm{Tony}, \mathrm{Bob})^{\mathsf{T}} \\
\mathrm{observes}(\mathrm{Tony}, \mathrm{murderer})^{\mathsf{T}} \\
\mathrm{black\_hair}(\mathrm{murderer})^{\mathsf{T}} \\
\neg\mathrm{black\_hair}(\mathrm{Bill})^{\mathsf{T}} \\
\mathrm{black\_hair}(\mathrm{Bill})^{\mathsf{F}} \\
\mathrm{black\_hair}(\mathrm{Bob})^{\mathsf{T}} \\
(\mathrm{observes}(\mathrm{Tony}, \mathrm{murderer}) \Rightarrow (\mathrm{murderer} = \mathrm{Bill} \lor \mathrm{murderer} = \mathrm{Bob}))^{\mathsf{T}} \\
(\mathrm{murderer} = \mathrm{Bill} \lor \mathrm{murderer} = \mathrm{Bob})^{\mathsf{T}} \\
\begin{array}{c|c}
\mathrm{murderer} = \mathrm{Bill}^{\mathsf{T}} & \mathrm{murderer} = \mathrm{Bob}^{\mathsf{T}} \\
\mathrm{black\_hair}(\mathrm{Bill})^{\mathsf{T}} & \\
\bot &
\end{array}
\end{array}$$

Rabbits [Gardent & Konrad '99]

- Interpret "the" as $\lambda PQ.Q\iota\ \ P\wedge \mathrm{uniq}(P)$
  where $\mathrm{uniq} := \lambda P.(\exists X.P(X)\wedge(\forall Y.P(Y)\Rightarrow X = Y))$
  and $\forall\!\!\!\forall := \lambda PQ.(\forall X.P(X)\Rightarrow Q(X))$.
- "the rabbit is cute", has logical form $\mathrm{uniq}(\mathrm{rabbit})\wedge(\mathrm{rabbit}\subseteq\mathrm{cute})$.
- *RM* generates $\{\ldots,\mathrm{rabbit}(c),\mathrm{cute}(c)\}$ in situations with at most 1 rabbit.
  (special *RM* $\exists$ rule yields identification and accommodation (c^{new}))

\+ At last an approach that takes world knowledge into account!

– tractable only for toy discourses/ontologies
  The world cup final was watched on TV by 7 million people.
  A rabbit is in the garden.
  $\forall X.\mathrm{human}(x)\exists Y.\mathrm{human}(X)\wedge\mathrm{father}(X,Y)\qquad\forall X,Y.\mathrm{father}(X,Y)\Rightarrow X\neq Y$

More than one Rabbit

- **Problem:** What about two rabbits?
  Bugs and Bunny are rabbits. Bugs is in the hat. Jon removes the rabbit from the hat.
- **Idea: Uniqueness under Scope [Gardent & Konrad '99]:**
  - refine *the* to $\lambda PRQ.\mathrm{uniq}(P \cap R \wedge \forall\!\!\!\forall(P \cap R, Q))$
    where R is an "identifying property" (identified from the context and passed as an arbument to *the*)
  - here R is "being in the hat" (by world knowledge about removing)
  - makes Bugs unique (in $P \cap R$) and the discourse acceptable.
- **Idea:** [Hobbs & Stickel&…]:
  - use generic relation rel for "relatedness to context" for P^2.

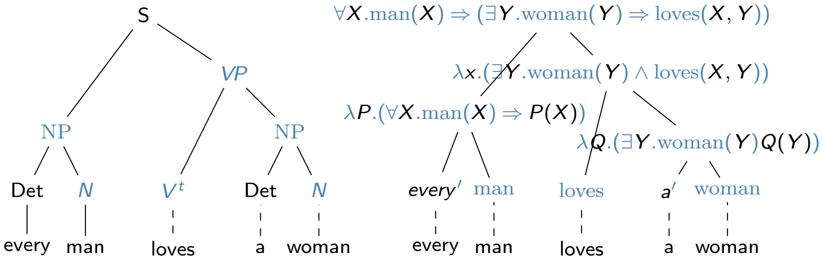
  ?? Is there a general theory of relatedness?

7.5 Quantifier Scope Ambiguity and Underspecification

7.5.1 Scope Ambiguity and Quantifying-In

Quantifier Scope Ambiguities: Data

- Consider the following sentences:
  1. *Every man loves a woman* (Britney Spears or his mother?)
  2. *Most Europeans speak two languages.*
  3. *Some student in every course sleeps in every class at least some of the time.*
- **Definition 5.1.** We call these systematic ambiguities quantifyer scope ambiguities
- **Example 5.2.** We can represent the "wide-scope" reading with our methods



- **Question:** How to map an unambiguous input structure to multiple translations.

Storing and Quantifying In

- **Analysis:** The sentence meaning is of the form $\langle \text{everyman} \rangle(\langle \text{awoman} \rangle(\langle \text{loves} \rangle))$
- **Idea:** Somehow have to move the *a woman* part in front of the *every* to obtain

$$\langle \text{awoman} \rangle(\langle \text{everyman} \rangle(\langle \text{loves} \rangle))$$

- **More concretely:** Let's try *A woman - every man loves her.*
  In semantics construction, apply *a woman* to *every man loves her*.
  So *a woman* out-scopes *every man*.
- **Problem:** How to represent pronouns and link them to their antecedents
- STORE is an alternative translation rule. Given a node with an NP daughter, we can translate the node by passing up to it the translation of its non-NP daughter, and putting the translation of the NP into a store, for later use.
- The QI rule allows us to empty out a non-empty store.

Storing and Quantifying In (Technically)

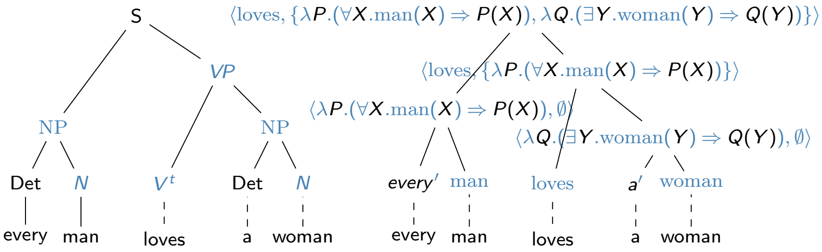
- **Definition 5.3.** $\mathrm{STORE}(NP,\Phi) \longrightarrow (\Phi,\Sigma{*}NP)$, where $\Sigma{*}NP$ is the result of adding NP to Σ, i.e. $\Sigma{*}NP=\Sigma\cup\{NP\}$; we will assume that NP is not already in Σ, when we use the $*$ operator.
- **Definition 5.4.** $\mathrm{QI}(\langle\Phi,\Sigma{*}NP\rangle) \rightarrow \langle NP\oplus\Phi,\Sigma\rangle$ where $\oplus$ is either function application or function composition.
- **Nondeterministic Semantics Construction:** Adding rules gives us more choice
  1. **Rule C (simple combination)** If A is a node with daughters B and C, and the translations of B and of C have empty stores, then A translates to $B'\oplus C'$. Choice of rule is determined by types.
  2. STORE If A is a node with daughters B and C, where:
     - B is an NP with translation B' and
     - C translates to (C',Σ)

     then A may translate to $\mathrm{STORE}(B',C')$

  Note that STORE may be applied whether or not the stores of the constituent nodes are empty.

Quantifying in Practice: *Every man loves a woman*

- **Example 5.5.**



- Continue with **QI** applications: first retrieve $\lambda Q.(\exists Y.\mathrm{woman}(Y) \Rightarrow Q(Y))$

$$\begin{array}{ll}
 & \langle \mathrm{loves}, \{\lambda P.(\forall X.\mathrm{man}(X) \Rightarrow P(X)), \lambda Q.(\exists Y.\mathrm{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{QI} & \langle \circ\,(\lambda P.(\forall X.\mathrm{man}(X) \Rightarrow P(X)))\ \mathrm{loves}, \{\lambda Q.(\exists Y.\mathrm{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{\beta} & \langle \lambda Z.(\lambda P.(\forall X.\mathrm{man}(X) \Rightarrow P(X)))\ \mathrm{loves}\ Z, \{\lambda Q.(\exists Y.\mathrm{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{\beta} & \langle \lambda Z.(\forall X.\mathrm{man}(X) \Rightarrow \mathrm{loves}\ Z\ X), \{\lambda Q.(\exists Y.\mathrm{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{QI} & \langle (\lambda Q.(\exists Y.\mathrm{woman}(Y) \Rightarrow Q(Y)))\ (\lambda Z.(\forall X.\mathrm{man}(X) \Rightarrow \mathrm{loves}\ Z\ X)), \emptyset\rangle \\
\rightarrow_{\beta} & \langle \exists Y.\mathrm{woman}(Y) \Rightarrow (\lambda Z.(\forall X.\mathrm{man}(X) \Rightarrow \mathrm{loves}\ Z\ X))\ Y, \emptyset\rangle \\
\rightarrow_{\beta} & \langle \exists Y.\mathrm{woman}(Y) \Rightarrow (\forall X.\mathrm{man}(X) \Rightarrow \mathrm{loves}\ Y\ X), \emptyset\rangle
\end{array}$$

7.5.2 Dealing with Quantifier Scope Ambiguity: Cooper Storage

Type raising transitive verbs

- We need transitive verbs to combine with quantificational objects of type $(\iota \to o) \to o$ but …
- We still ultimately want their "basic" translation to be type $\iota \to \iota \to o$, i.e. something that denotes a relation between individuals.
- We do this by starting with the basic translation, and raising its type. Here is what we'll end up with, for the verb *like*:

$$\lambda PY.P\,(\lambda X.\mathrm{likes}(X, Y))$$

where P is a variable of type $(\iota \to o) \to o$ and X, Y are variables of type ι. (For details on how this is derived, see [CKG09, pp.178-179])

Cooper Storage

- **Intuition:** A store consists of a "core" semantic representation, computed in the usual way, plus the representations of quantifiers encountered in the composition so far.
- **Definition 5.6.** A store is an n place sequence. The first member of the sequence is the core semantic representation. The other members of the sequence (if any) are pairs (β, i) where:
  - β is a QNP translation and
  - i is an index, which will associate the NP translation with a free variable in the core semantic translation.

  We call these pairs binding operators (because we will use them to bind free variables in the core representation).
- **Definition 5.7.** In the Cooper storage method, QNPs are stored in the store and later retrieved – not necessarily in the order they were stored – to build the representation.
- The elements in the store are written enclosed in angled brackets. However, we will often have a store which consists of only one element, the core semantic representation. This is because QNPs are the only things which add elements beyond the core representation to the store. So we will adopt the convention that when the store has only one element, the brackets are omitted.

How we put QNPs in the Store

- **Storage Rule**
  If the store $\langle\varphi, (\beta, j), \ldots, (\gamma, k)\rangle$ is a possible translation for a QNP, then the store
  $$\langle\lambda P.P(X_i)(\varphi, i)(\beta, j), \ldots, (\gamma, k)\rangle$$
  where i is a new index, is also a possible translation for that QNP.
- This rule says: if you encounter a QNP with translation φ, you can replace its translation with an indexed place holder of the same type, $\lambda P.P(X_i)$, and add φ to the store, paired with the index i. We will use the place holder translation in the semantic composition of the sentence.

Working with Stores

- Working out the translation for *Every student likes some professor.*

$$
\begin{array}{lll}
NP_1 & \rightarrow & \lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)) \textbf{ or } \langle \lambda Q.Q(X_1), (\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)), 1)\rangle \\
V_t & \rightarrow & \lambda RY.R\ (\lambda Z.\mathrm{likes}(Z,Y)) \\
VP & \rightarrow & \text{(Combine core representations by FA; pass store up)*} \\
 & \rightarrow & \langle \lambda Y.\mathrm{likes}(X_1,Y), (\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)), 1)\rangle \\
NP_2 & \rightarrow & \lambda P.(\forall Z.\mathrm{student}(Z) \Rightarrow P(Z)) \textbf{ or } \langle \lambda R.R(X_2), (\lambda P.(\forall Z.\mathrm{student}(Z) \Rightarrow P(Z)), 2 \\
S & \rightarrow & \text{(Combine core representations by FA; pass stores up)**} \\
 & \rightarrow & \langle \mathrm{likes}(X_1,X_2), (\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)), 1), (\lambda P.(\forall Z.\mathrm{student}(Z) \Rightarrow P(Z)), 2
\end{array}
$$

* Combining V_t with place holder

1. $(\lambda RY.R\ (\lambda Z.\mathrm{likes}(Z,Y)))\ (\lambda Q.Q(X_1))$
2. $\lambda Y.(\lambda Q.Q(X_1))\ (\lambda Z.\mathrm{likes}(Z,Y))$
3. $\lambda Y.(\lambda Z.\mathrm{likes}(Z,Y))\ X_1$
4. $\lambda Y.\mathrm{likes}(X_1,Y)$

** Combining VP with place holder

1. $(\lambda R.R(X_2))\ (\lambda Y.\mathrm{likes}(X_1,Y))$
2. $(\lambda Y.\mathrm{likes}(X_1,Y))\ X_2$
3. $\mathrm{likes}(X_1,X_2)$

Retrieving NPs from the store

- **Retrieval:**
  Let σ_1 and σ_2 be (possibly empty) sequences of binding operators. If the store $\langle\varphi, \sigma_1, \sigma_2, (\beta, i)\rangle$ is a translation of an expression of category S, then the store $\langle\beta(\lambda X_1.\varphi), \sigma_1, \sigma_2\rangle$ is also a translation of it.
- **What does this say?:** It says: suppose you have an S translation consisting of a core representation (which will be of type o) and one or more indexed QNP translations. Then you can do the following:
  1. Choose one of the QNP translations to retrieve.
  2. Rewrite the core translation, λ-abstracting over the variable which bears the index of the QNP you have selected. (Now you will have an expression of type $\iota \to o$.)
  3. Apply this λ-term to the QNP translation (which is of type $(\iota \to o) \to o$).

Example: *Every student likes some professor.*

1. Retrieve *every student*
   1.1 $(\lambda Q.(\forall Z.\mathrm{student}(Z) \Rightarrow Q(Z)))\;(\lambda X_2.\mathrm{likes}(X_1, X_2))$
   1.2 $\forall Z.\mathrm{student}(Z) \Rightarrow (\lambda X_2.\mathrm{likes}(X_1, X_2))\; Z$
   1.3 $\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X_1, Z)$
2. Retrieve *some professor*
   2.1 $(\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)))\;(\lambda X_1.(\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X_1, Z)))$
   2.2 $\exists X.\mathrm{prof}(X)(\lambda X_1.(\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X_1, Z)))\; X$
   2.3 $\exists X.\mathrm{prof}(X) \wedge (\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X, Z))$

7.6 Summary & Evaluation

Fragment $\mathcal{F}_4$ – Summary

- Fragment $\mathcal{F}_4$ extends $\mathcal{F}_3$ by noun phrases.
- **Coverage:** Better:

Chapter 8
Davidsonian Semantics: Treating Verb Modifiers

Event semantics: Davidsonian Systems

- **Problem:** How to deal with argument structure of (action) verbs and their modifiers
  - *John killed a cat with a hammer.*
- **Idea:** Just add an argument to kills for express the means
- **Problem:** But there may be more modifiers

  1. *Peter killed the cat in the bathroom with a hammer.*
  2. *Peter killed the cat in the bathroom with a hammer at midnight.*

  So we would need a lot of different predicates for the verb *killed*. (impractical)
- **Definition 0.1.** In event semantics we extend the argument structure of (action) verbs contains a 'hidden' argument, the event argument, then treat modifiers as predicates (often called roles) over events [Dav67a].
- **Example 0.2.**
  1. $\exists e.\exists x, y.\mathrm{bathroom}(x) \wedge \mathrm{hammer}(y) \wedge \mathrm{kill}(e, \mathrm{peter}, \iota\ \mathrm{cat}) \wedge \mathrm{in}(e, x) \wedge \mathrm{with}(e, y)$
  2. $\exists e.\exists x, y.\mathrm{bathroom}(x) \wedge \mathrm{hammer}(y) \wedge \mathrm{kill}(e, \mathrm{peter}, \iota\ \mathrm{cat}) \wedge \mathrm{in}(e, x) \wedge \mathrm{with}(e, y) \wedge \mathrm{at}(e, 24:00)$

Event semantics: Neo-Davidsonian Systems

- **Idea:** Take apart the Davidsonian predicates even further, add event participants via thematic roles (from [Par90]).
- **Definition 0.3.** Neo-Davidsonian semantics extends event semantics by adding two standardized roles: the agent $\mathrm{ag}(e,s)$ and the patient $\mathrm{pat}(e,o)$ for the subject s and direct object d of the event e.
- **Example 0.4.** Translate *John killed a cat with a hammer.* as $\exists e.\exists x.\mathrm{hammer}(x) \wedge \mathrm{killing}(e) \wedge \mathrm{ag}(e,\mathrm{peter}) \wedge \mathrm{pat}(e,\iota\ \mathrm{cat}) \wedge \mathrm{with}(e,x)$
- **Further Elaboration:** Events can be broken down into sub-events and modifiers can predicate over sub-events.
- **Example 0.5.** The "process" of climbing Mt. Everest starts with the "event" of (optimistically) leaving the base camp and culminates with the "achievement" of reaching the summit (being completely exhausted).
- **Note:** This system can get by without functions, and only needs unary and binary predicates. (well-suited for model generation)

Event Types and Properties of Events

- **Example 0.6 (Problem).** Some (temporal) modifiers are incompatible with some events, e.g. in English progressive:
  1. *He is eating a sandwich* and *He is pushing the cart.*, but not
  2. * *He is being tall.* or * *He is finding a coin.*
- **Definition 0.7 (Types of Events).** There are different types of events that go with different temporal modifiers. [Ven57] distinguishes
  1. states: e.g. *know the answer*, *stand in the corner*
  2. processes: e.g. *run*, *eat*, *eat apples*, *eat soup*
  3. accomplishments: e.g. *run a mile*, *eat an apple*, and
  4. achievements: e.g. *reach the summit*
- **Observations:**
  1. processes and accomplishments appear in the progressive (1),
  2. states and achievements do not (2).
- **Definition 0.8.** The in test
  1. states and activities, but not accomplishments and achievements are compatible with *for*-adverbials
  2. whereas the opposite holds for in-adverbials (5).
- **Example 0.9.**
  1. *run a mile in an hour* vs. * *run a mile for an hour*, but
  2. * *reach the summit for an hour* vs *reach the summit in an hour*

Part 2
Topics in Semantics

Chapter 9
Dynamic Approaches to NL Semantics

9.1 Discourse Representation Theory

Anaphora and Indefinites revisited (Data)

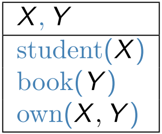
- **Observation:** We have concentrated on single sentences so far; let's do better.
- **Definition 1.1.** A discourse is a unit of natural language longer than a single sentence.
- **New Data:** Discourses interact with anaphora.:
  - *Peter^1 is sleeping. He$_1$ is snoring.* (normal anaphoric reference)
  - *A man^1 is sleeping. He$_1$ is snoring.* (scope of existential?)
  - *Peter has a car^1. It$_1$ is parked outside.* (even if this worked)
  - * *Peter has no car^1. It$_1$ is parked outside.* (what about negation?)
  - *There is a book^1 that Peter does not own. It$_1$ is a novel.* (OK)
  - * *Peter does not own every book^1. It$_1$ is a novel.* (equivalent in PL^1)
  - *If a farmer^1 owns a donkey$_2$, he$_1$ beats it$_2$.* (even inside sentences)
- We gloss the intended anaphoric reference with the labels in upper and lower indices.

Dynamic Effects in Natural Language

- **Problem:** E.g. Quantifier Scope
  - * *A man sleeps. He snores.*
  - $(\exists X.\mathrm{man}(X) \wedge \mathrm{sleeps}(X)) \wedge \mathrm{snores}(X)$
  - X is bound in the first conjunct, and free in the second.
- **Problem:** Donkey sentence: *If a farmer owns a donkey, he beats it.* $\forall X, Y.\mathrm{farmer}(X) \wedge \mathrm{donkey}(Y) \wedge \mathrm{own}(X, Y) \Rightarrow \mathrm{beat}(X, Y)$
- **Ideas:**
  - Composition of sentences by conjunction inside the scope of existential quantifiers (non-compositional, ...)
  - Extend the scope of quantifiers dynamically (DPL)
  - Replace existential quantifiers by something else (DRT)

Discourse Representation Theory (DRT)

- **Definition 1.2.** Discourse Representation Theory (DRT) is a logical system, which uses discourse referents to model quantification and pronouns. DRT formulae are called discourse representation structures (DRS); these introduce a set of discourse referents and specify their meaning by conditions which comprise:
  - atomic first-order propositions,
  - dynamic negations $\neg D$,
  - dynamic implications $D \Rrightarrow E$, and
  - dynamic disjunctions $D \veebar E$.
- **Example 1.3.** Discourse referents e.g. in *A student owns a book.*
  - are kept in a dynamic context (⇝ accessibility)
  - are declared e.g. in indefinite nominals
  - specified in conditions via predicates

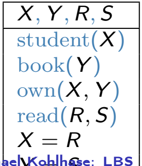


X, Y

student(X)
book(Y)
own(X, Y)

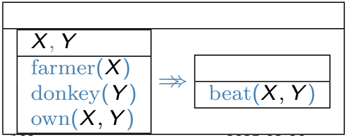
- **Example 1.4.** Discourse representation structures (DRS)

A student owns a book. He reads it. *If a farmer owns a donkey, he beats it.*



X, Y, R, S

student(X)
book(Y)
own(X, Y)
read(R, S)
$X = R$
$Y = S$



X, Y

farmer(X)
donkey(Y)
own(X, Y)

$\Rrightarrow$

beat(X, Y)

Discourse DRS Construction

- **Problem:** How do we construct DRSes for multi-sentence discourses?
- **Solution:** We construct sentence DRSes individually and merge them (DRSes and conditions separately)
- **Example 1.5.** A three-sentence discourse. (not quite Shakespeare)

Mary sees John.

see(mary, john)

John kills a cat.

U
cat(U) kills(john, U)

Mary calls a cop.

V
policeman(V) calls(mary, V)

merge

U, V
see(mary, john) cat(U) kills(john, U) policeman(V) calls(mary, V)

- Sentence composition via the DRT Merge Operator $\otimes$. (acts on DRSes)

Anaphor Resolution in DRT

- **Problem:** How do we resolve anaphora in DRT?
- **Solution:** Two phases
  - translate pronouns into discourse referents (semantics construction)
  - identify (equate) coreferring discourse referents, (maybe) simplify (semantic/pragmatic analysis)
- **Example 1.6.** *A student owns a book. He reads it.*

A student^1 owns a book^2.

$$\begin{array}{|l|}\hline X, Y \\ \hline \mathrm{student}(X) \\ \mathrm{book}(Y) \\ \mathrm{own}(X,Y) \\ \hline \end{array}$$

He$_1$ reads it$_2$

$$\begin{array}{|l|}\hline R, S \\ \hline \mathrm{read}(R,S) \\ \hline \end{array}$$

merge/resolve

$$\begin{array}{|l|}\hline X, Y, R, S \\ \hline \mathrm{student}(X) \\ \mathrm{book}(Y) \\ \mathrm{own}(X,Y) \\ \mathrm{read}(R,S) \\ X = R \\ Y = S \\ \hline \end{array}$$

simplify

$$\begin{array}{|l|}\hline X, Y \\ \hline \mathrm{student}(X) \\ \mathrm{book}(Y) \\ \mathrm{own}(X,Y) \\ \mathrm{read}(X,Y) \\ \hline \end{array}$$

DRT (more Logic-like Syntax)

- **Definition 1.7.** Given a set $\mathcal{DR}$ of discourse referents, discourse representation structure (DRSes) are given by the following grammar:

  conditions $\mathcal{C} ::= p(a_1, \ldots, a_n) \mid \mathcal{C}_1 \wedge \mathcal{C}_2 \mid \neg\mathcal{D} \mid \mathcal{D}_1 \mathbb{V} \mathcal{D}_2 \mid \mathcal{D}_1 \Rrightarrow \mathcal{D}_2$

  DRSes $\mathcal{D} ::= \delta U^1, \ldots, U^n.\mathcal{C} \mid \mathcal{D}_1 \otimes \mathcal{D}_2 \mid \mathcal{D}_1 \,;;\, \mathcal{D}_2$

- $\otimes$ and ;; are for sentence composition (⊗ from DRT, ;; from DPL)
- **Example 1.8.** $\delta U, V.\mathrm{farmer}(U) \wedge \mathrm{donkey}(V) \wedge \mathrm{own}(U, V) \wedge \mathrm{beat}(U, V)$
- **Definition 1.9.** The meaning of $\otimes$ and ;; is given operationally by $=_\tau$ equality:

$$\begin{aligned} \delta\mathcal{X}.\mathcal{C}_1 \otimes \delta\mathcal{Y}.\mathcal{C}_2 &\to_\tau \delta\mathcal{X}, \mathcal{Y}.\mathcal{C}_1 \wedge \mathcal{C}_2 \\ \delta\mathcal{X}.\mathcal{C}_1 \,;;\, \delta\mathcal{Y}.\mathcal{C}_2 &\to_\tau \delta\mathcal{X}, \mathcal{Y}.\mathcal{C}_1 \wedge \mathcal{C}_2 \end{aligned}$$

- Discourse referents used instead of bound variables. (specify scoping independently of logic)
- **Idea:** Semantics inherited from first-order logic by a translation mapping.

Sub DRSes and Accessibility

- **Problem:** How can we formally define accessibility. (to make predictions)
- **Idea:** Make use of the structural properties of DRT.
- **Definition 1.10.** A referent is accessible in all sub DRS of the declaring DRS.
  - If $\mathcal{D} = \delta U^1, \ldots, U^n.\mathcal{C}$, then any sub DRS of $\mathcal{C}$ is a sub DRS of $\mathcal{D}$.
  - If $\mathcal{D} = \mathcal{D}^1 \otimes \mathcal{D}^2$, then $\mathcal{D}^1$ is a sub DRS of $\mathcal{D}^2$ and vice versa.
  - If $\mathcal{D} = \mathcal{D}^1 ;; \mathcal{D}^2$, then $\mathcal{D}^2$ is a sub DRS of $\mathcal{D}^1$.
  - If $\mathcal{C}$ is of the form $\mathcal{C}^1 \wedge \mathcal{C}^2$, or $\neg\mathcal{D}$, or $\mathcal{D}^1 \Vvee \mathcal{D}^2$, or $\mathcal{D}^1 \twoheadRightarrow \mathcal{D}^2$, then any sub DRS of the $\mathcal{C}^i$, and the $\mathcal{D}^i$ is a sub DRS of $\mathcal{C}$.
  - If $\mathcal{D} = \mathcal{D}^1 \twoheadRightarrow \mathcal{D}^2$, then $\mathcal{D}^2$ is a sub DRS of $\mathcal{D}^1$
- **Definition 1.11 (Dynamic Potential).** (which referents can be picked up?) A referent U is in the dynamic potential of a DRS $\mathcal{D}$, iff it is accessible in

  $$\mathcal{D} \otimes \begin{array}{|c|}\hline \\ \hline p(U) \\ \hline\end{array}$$
- **Definition 1.12.** We call a DRS static, iff the dynamic potential is empty, and dynamic, if it is not.

Sub DRSes and Accessibility

- **Observation:** Accessibility gives DRSes the flavor of binding structures. (with non-standard scoping!)
- **Idea:** Apply the usual binding heuristics to DRT, e.g.
  - reject DRSes with unbound discourse referents.
- **Questions:** If we view of discourse referents as “nonstandard bound variables”
  - what about renaming referents?

Translation from DRT to FOL

- **Definition 1.13.** For $=_\tau$-normal (fully merged) DRSes use the translation $\overline{\cdot}$:

$$\begin{aligned}
\overline{\delta U^1, \ldots, U^n.\mathcal{C}} &= \exists U^1, \ldots, U^n.\overline{\mathcal{C}} \\
\overline{\neg \mathcal{D}} &= \neg \overline{\mathcal{D}} \\
\overline{\mathcal{D} \mathbb{V} \mathcal{E}} &= \overline{\mathcal{D}} \vee \overline{\mathcal{E}} \\
\overline{\mathcal{D} \wedge \mathcal{E}} &= \overline{\mathcal{D}} \wedge \overline{\mathcal{E}} \\
\overline{(\delta U^1, \ldots, U^n.\mathcal{C}_1) \Rrightarrow (\delta V^1, \ldots, V^n.\mathcal{C}_2)} &= \forall U^1, \ldots, U^n.\overline{\mathcal{C}_1} \Rightarrow (\exists V^1, \ldots, V^n.\overline{\mathcal{C}_2})
\end{aligned}$$

- **Example 1.14.** $\overline{\begin{array}{|l|}\hline X, Y \\ \hline \text{student}(X) \\ \text{book}(Y) \\ \text{own}(X, Y) \\ \hline \end{array}} = \exists X.\exists Y.\text{student}(X) \wedge \text{book}(Y) \wedge \text{own}(X, Y)$.

- **Example 1.15.**

$$\overline{(\delta U, V.\text{farmer}(U) \wedge \text{donkey}(V) \wedge \text{own}(U, V)) \Rrightarrow (\delta W.\text{stick}(W) \wedge \text{beatwith}(U, V, W))}$$
$$= \forall X, Y.\text{farmer}(X) \wedge \text{donkey}(X) \wedge \text{own}(X, Y) \Rightarrow (\exists.\text{stick}(Z) \wedge \text{beatwith}(Z, X, Y))$$

- **Consequence:** Validity of DRSes can be checked by translation.
- **Question:** Why not use first-order logic directly?
- **Answer:** Only translate at the end of a discourse(translation closes all dynamic contexts: frequent re-translation).

Properties of Dynamic Scope

- **Idea:** Test DRT on the data above for the dynamic phenomena
- **Example 1.16 (Negation Closes Dynamic Potential).**

  Peter has no^1 car. * *It$_1$ is parked outside.*

$$\begin{array}{|l|}\hline \\ \hline \neg \begin{array}{|l|}\hline U \\ \hline \mathrm{acar}(U) \\ \mathrm{own}(\mathrm{peter}, U) \\ \hline \end{array} \\ \hline \end{array} \otimes \begin{array}{|l|}\hline \\ \hline \mathrm{parked}(U) \\ \hline \end{array} \qquad \neg(\exists U.\mathrm{acar}(U) \wedge \mathrm{own}(\mathrm{peter}, U))\ldots$$

- **Example 1.17 (Universal Quantification is Static).**

  Peter does not own every book^1. * *It$_1$ is a novel.*

$$\begin{array}{|l|}\hline \\ \hline \neg \begin{array}{|l|}\hline \\ \hline \begin{array}{|l|}\hline U \\ \hline \mathrm{book}(U) \\ \hline \end{array} \Rrightarrow \begin{array}{|l|}\hline \\ \hline \mathrm{own}(\mathrm{peter}, U) \\ \hline \end{array} \\ \hline \end{array} \\ \hline \end{array} \otimes \begin{array}{|l|}\hline \\ \hline \mathrm{novel}(U) \\ \hline \end{array}$$

$$\neg(\forall U.\mathrm{book}(U) \Rightarrow \mathrm{own}(\mathrm{peter}, U))\ldots$$

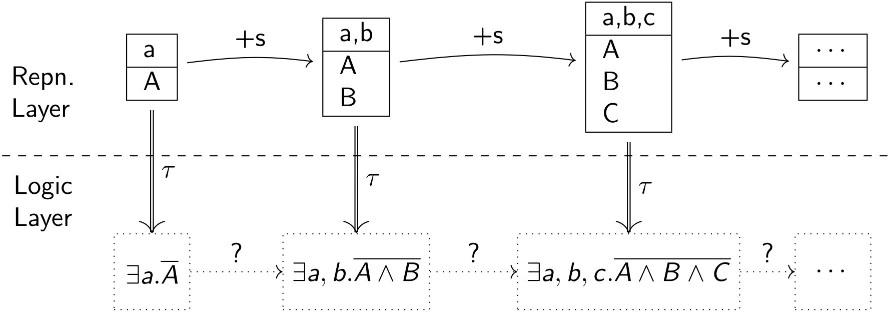
- **Example 1.18 (Existential Quantification is Dynamic).**

  There is a book^1 that Peter does not own. *It$_1$ is a novel.*

$$\begin{array}{|l|}\hline V \\ \hline \mathrm{book}(V) \\ (\neg\mathrm{own}(\mathrm{peter}, V)) \\ \hline \end{array} \otimes \begin{array}{|l|}\hline \\ \hline \mathrm{novel}(V) \\ \hline \end{array} \qquad \exists U.\mathrm{book}(U) \wedge \neg\mathrm{own}(\mathrm{peter}, U) \wedge \mathrm{novel}(U)$$

DRT as a Representational Level

- DRT adds a level to the knowledge representation which provides anchors (the discourse referents) for anaphora and the like.
- Propositional semantics by translation into PL^1. ("+s" adds a sentence)



Repn. Layer

a
A

+s

a,b
A B

+s

a,b,c
A B C

+s

...
...

τ τ τ

Logic Layer

$\exists a.\overline{A}$? $\exists a,b.\overline{A \wedge B}$? $\exists a,b,c.\overline{A \wedge B \wedge C}$? $\cdots$

- Anaphor resolution works incrementally on the representational level.

A Direct Semantics for DRT (Dyn. Interpretation $\mathcal{I}_\varphi^\delta$)

- **Definition 1.19.** Let $\mathcal{M} = \langle\mathcal{D},\mathcal{I}\rangle$ be a first-order model, then a state is an assignment from discourse referents into $\mathcal{D}$.
- **Definition 1.20.** Let $\varphi, \psi : \mathcal{DR} \to \mathcal{U}$ be states, then we say that ψ extends φ on $\mathcal{X} \subseteq \mathcal{DR}$ (write $\varphi[\mathcal{X}]\psi$), if $\varphi(U) = \psi(U)$ for all $U \notin \mathcal{X}$.
- **Idea:** Conditions as truth values; DRSes as pairs $(\mathcal{X},\mathcal{S})$ ($\mathcal{S}$ set of states)
- **Definition 1.21 (Meaning of complex formulae).** The value function $\mathcal{I}_\varphi$ for DRT is defined with the help of a dynamic value function $\mathcal{I}_\varphi^\delta$ on DRSs: For conditions:
  - $\mathcal{I}_\varphi(\neg\mathcal{D}) = \mathsf{T}$, if $\mathcal{I}_\varphi^\delta(\mathcal{D})^2 = \emptyset$.
  - $\mathcal{I}_\varphi(\mathcal{D}\Cup\mathcal{E}) = \mathsf{T}$, if $\mathcal{I}_\varphi^\delta(\mathcal{D})^2 \neq \emptyset$ or $\mathcal{I}_\varphi^\delta(\mathcal{E})^2 \neq \emptyset$.
  - $\mathcal{I}_\varphi(\mathcal{D}\Rrightarrow\mathcal{E}) = \mathsf{T}$, if for all $\psi \in \mathcal{I}_\varphi^\delta(\mathcal{D})^2$ there is a $\tau \in \mathcal{I}_\varphi^\delta(\mathcal{E})^2$ with $\psi[\mathcal{I}_\varphi^\delta(\mathcal{E})^1]\tau$.

  For DRSs $\mathcal{D}$ we set $\mathcal{I}_\varphi(\mathcal{D}) = \mathsf{T}$, iff $\mathcal{I}_\varphi^\delta(\mathcal{D})^2 \neq \emptyset$, and define

  - $\mathcal{I}_\varphi^\delta(\delta\mathcal{X}.\mathsf{C}) = (\mathcal{X},\{\psi \mid \varphi[\mathcal{X}]\psi \text{ and } \mathcal{I}_\psi(\mathsf{C}) = \mathsf{T}\})$.
  - $\mathcal{I}_\varphi^\delta(\mathcal{D} \otimes \mathcal{E}) = \mathcal{I}_\varphi^\delta(\mathcal{D} \mathbin{;;} \mathcal{E}) = (\mathcal{I}_\varphi^\delta(\mathcal{D})^1 \cup \mathcal{I}_\varphi^\delta(\mathcal{E})^1, \mathcal{I}_\varphi^\delta(\mathcal{D})^2 \cap \mathcal{I}_\varphi^\delta(\mathcal{E})^2)$

Examples (Computing Direct Semantics)

- **Example 1.22.** *Peter owns a car*

$$\begin{aligned}
& \mathcal{I}^{\delta}_{\varphi}(\delta U.\mathrm{acar}(U) \wedge \mathrm{own}(\mathrm{peter}, U)) \\
= \quad & (\{U\},\{\psi \mid \varphi[U]\psi \text{ and } \mathcal{I}_{\psi}(\mathrm{acar}(U) \wedge \mathrm{own}(\mathrm{peter}, U)) = \mathsf{T}\}) \\
= \quad & (\{U\},\{\psi \mid \varphi[U]\psi \text{ and } \mathcal{I}_{\psi}(\mathrm{acar}(U)) = \mathsf{T} \text{ and } \mathcal{I}_{\psi}(\mathrm{own}(\mathrm{peter}, U)) = \mathsf{T}\}) \\
= \quad & (\{U\},\{\psi \mid \varphi[U]\psi \text{ and } \psi(U) \in \mathcal{I}(\mathrm{acar}) \text{ and } (\psi(U),\mathrm{peter}) \in \mathcal{I}(\mathrm{own})\})
\end{aligned}$$

The set of states $[a/U]$, such that a is a car and is owned by Peter

- **Example 1.23.** For *Peter owns no car* we look at the condition:

$$\begin{aligned}
& \mathcal{I}_{\varphi}(\neg(\delta U.\mathrm{acar}(U) \wedge \mathrm{own}(\mathrm{peter}, U))) = \mathsf{T} \\
\Leftrightarrow \quad & \mathcal{I}^{\delta}_{\varphi}(\delta U.\mathrm{acar}(U) \wedge \mathrm{own}(\mathrm{peter}, U))^{2} = \emptyset \\
\Leftrightarrow \quad & (\{U\},\{\psi \mid \varphi[\mathcal{X}]\psi \text{ and } \psi(U) \in \mathcal{I}(\mathrm{acar}) \text{ and } (\psi(U),\mathrm{peter}) \in \mathcal{I}(\mathrm{own})\})^{2} = \emptyset \\
\Leftrightarrow \quad & \{\psi \mid \varphi[\mathcal{X}]\psi \text{ and } \psi(U) \in \mathcal{I}(\mathrm{acar}) \text{ and } (\psi(U),\mathrm{peter}) \in \mathcal{I}(\mathrm{own})\} = \emptyset
\end{aligned}$$

i.e. iff there are no a, that are cars and that are owned by Peter.

9.2 Dynamic Model Generation

Deduction in Dynamic Logics

- **Problem:** Mechanize the dynamic entailment relation (with anaphora)
- **Idea:** Use dynamic deduction theorem to reduce (dynamic) entailment to (dynamic) satisfiability
- **History of Attempts:** Direct Deduction on DRT (or DPL) [Sau93; RG94; MR98]
  - (++) Specialized Calculi for dynamic representations.
  - (− −) Needs lots of development until we have efficient implementations.
- Translation approach (used in our experiment)
  - (−) Translate to PL^1.
  - (++) Use off-the-shelf theorem prover (in this case `MathWeb`).

An Opportunity for Off-The-Shelf ATP?

- **Pro:** ATP is mature enough to tackle applications
  - Current ATP are highly efficient reasoning tools.
  - Full automation is needed for NLP. (ATP as an oracle)
  - ATP as logic engines is one of the initial promises of the field.
- **contra:** ATP are general logic systems
  1. NLP uses other representation formalisms (DRT, Feature Logic,. . .)
  2. ATP optimized for mathematical (combinatorially complex) proofs.
  3. ATP (often) do not terminate.
- **Experiment:** Use translation approach for 1. to test 2. and 3. [Bla+01] (Wow, it works!)

Excursion: Incrementality in Dynamic Calculi

- For applications, we need to be able to check for
  - satisfiability ($\exists\mathcal{M}.\mathcal{M}\models A$), validity ($\forall\mathcal{M}.\mathcal{M}\models A$) and
  - entailment ($\mathcal{H}\models A$, iff $\mathcal{M}\models\mathcal{H}$ implies $\mathcal{M}\models A$ for all $\mathcal{M}$)
- **Theorem 2.1 (Entailment Theorem).** *$\mathcal{H}, A\models B$, iff $\mathcal{H}\models A\Rightarrow B$.* *(e.g. for first-order logic and DPL)*
- **Theorem 2.2 (Deduction Theorem).** *For most calculi $\mathcal{C}$ we have $\mathcal{H}, A\vdash_{\mathcal{C}}B$, iff $\mathcal{H}\vdash_{\mathcal{C}}A\Rightarrow B$.* *(e.g. for $\mathcal{ND}^1$)*
- **Problem:** Analogue $H_1\otimes\cdots\otimes H_n\models A$ is not equivalent to $\models(H_1\otimes\cdots\otimes H_n)\Rrightarrow A$ in DRT, as $\otimes$ symmetric.
- **Thus** the validity check cannot be used for entailment in DRT.
- **Solution:** Use sequential merge ;; (from DPL) for sentence composition.

Model Generation for Dynamic Logics

- **Problem:** Translation approach is not incremental!
  - For each check, the DRS for the whole discourse has to be translated.
  - Can become infeasible, once discourses get large (e.g. novel).
  - This applies for all other approaches for dynamic deduction too.
- **Idea:** Extend model generation techniques instead!
  - **Remember**: A DRS $\mathcal{D}$ is valid in $\mathcal{M} = \langle\mathcal{D},\mathcal{I}\rangle$, iff $\mathcal{I}_{\emptyset}^{\delta}(\mathcal{D})^2 \neq \emptyset$.
  - Find a model $\mathcal{M}$ and state φ, such that $\varphi \in \mathcal{I}_{\emptyset}^{\delta}(\mathcal{D})^2$.
  - Adapt first-order model generation technology for that.

Dynamic Herbrand Valuation

- **Definition 2.3.** We call a model $\mathcal{M} = \langle \mathcal{U}, \mathcal{I}, \mathcal{I}^{\delta}_{\cdot} \rangle$ a dynamic Herbrand interpretation, if $\langle \mathcal{U}, \mathcal{I} \rangle$ is a Herbrand model.
- **Question:** Can represent $\mathcal{M}$ as a triple $\langle \mathcal{X}, \mathcal{S}, \mathcal{B} \rangle$, where $\mathcal{B}$ is the Herbrand valuation for $\langle \mathcal{U}, \mathcal{I} \rangle$?
- **Definition 2.4.** $\mathcal{M}$ is called finite, iff $\mathcal{U}$ is finite.
- **Definition 2.5.** $\mathcal{M}$ is minimal, iff for all $\mathcal{M}'$ the following holds: $(\mathcal{B}(\mathcal{M})' \subseteq \mathcal{B}(\mathcal{M})) \Rightarrow \mathcal{M} = \mathcal{M}'$.
- **Definition 2.6.** $\mathcal{M}$ is domain minimal if for all $\mathcal{M}'$ the following holds:

$$\#(\mathcal{U}(\mathcal{M})) \leq \#(\mathcal{U}(\mathcal{M})')$$

Dynamic Model Generation Calculus

- **Definition 2.7.** We use a tableau framework, extend by state information, and rules for DRSes.
  - $$\frac{(\delta U_{\mathbb{A}}.\mathrm{A})^{\mathsf{T}} \quad \mathcal{H}=\{a_1,\ldots,a_n\} \quad w\notin\mathcal{H}\ \text{new}}{\begin{array}{c|c|c|c} [a_1/U] & & [a_n/U] & [w/U] \\ ([a_1/U](\mathrm{A}))^{\mathsf{T}} & \ldots & ([a_n/U](\mathrm{A}))^{\mathsf{T}} & ([w/U](\mathrm{A}))^{\mathsf{T}} \end{array}}\ RM\,\delta$$
  - Mechanize ;; by adding representation of the second DRS at all leaves. ($\leadsto$ tableau machine)
  - Treat conditions by DRT translation

$$\frac{\neg\mathcal{D}}{\overline{\neg\mathcal{D}}} \qquad \frac{\mathcal{D}\Rrightarrow\mathcal{D}'}{\overline{\mathcal{D}\Rrightarrow\mathcal{D}'}} \qquad \frac{\mathcal{D}\mathbb{V}\mathcal{D}'}{\overline{\mathcal{D}\mathbb{V}\mathcal{D}'}}$$

Example: *Peter is a man. No man walks*

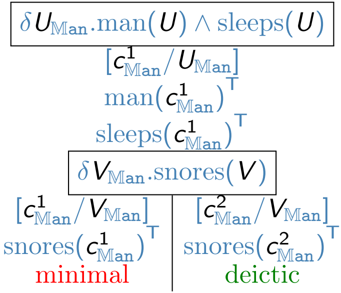
- **Example 2.8 (Model Generation).** *Peter is a man. No man walks*

$$\begin{array}{c}
\boxed{\mathrm{man}(\mathrm{peter})}\\
\mathrm{man}(\mathrm{peter})^{\mathsf{T}}\\
\boxed{\neg(\delta U.\mathrm{man}(U)\wedge\mathrm{walks}(U))}\\
\neg(\forall U.\mathrm{man}(U)\wedge\mathrm{walks}(U))^{\mathsf{T}}\\
(\forall X.\mathrm{man}(X)\wedge\mathrm{walks}(X))^{\mathsf{F}}\\
(\mathrm{man}(\mathrm{peter})\wedge\mathrm{walks}(\mathrm{peter}))^{\mathsf{F}}\\
\begin{array}{c|c}
\mathrm{man}(\mathrm{peter})^{\mathsf{F}} & \mathrm{walks}(\mathrm{peter})^{\mathsf{F}}\\
\bot &
\end{array}
\end{array}$$

Dynamic Herbrand interpretation: $\langle\emptyset,\emptyset,\{\mathrm{man}(\mathrm{peter})^{\mathsf{T}},\mathrm{walks}(\mathrm{peter})^{\mathsf{F}}\}\rangle$

Example: Anaphor Resolution *A man sleeps. He snores*

- **Example 2.9 (Anaphor Resolution).** *A man sleeps. He snores*



$$\boxed{\delta U_{\mathbb{M}\mathrm{an}}.\mathrm{man}(U) \wedge \mathrm{sleeps}(U)}$$

$$[c^1_{\mathbb{M}\mathrm{an}}/U_{\mathbb{M}\mathrm{an}}]$$

$$\mathrm{man}(c^1_{\mathbb{M}\mathrm{an}})^{\mathsf{T}}$$

$$\mathrm{sleeps}(c^1_{\mathbb{M}\mathrm{an}})^{\mathsf{T}}$$

$$\boxed{\delta V_{\mathbb{M}\mathrm{an}}.\mathrm{snores}(V)}$$

$[c^1_{\mathbb{M}\mathrm{an}}/V_{\mathbb{M}\mathrm{an}}]$	$[c^2_{\mathbb{M}\mathrm{an}}/V_{\mathbb{M}\mathrm{an}}]$
$\mathrm{snores}(c^1_{\mathbb{M}\mathrm{an}})^{\mathsf{T}}$	$\mathrm{snores}(c^2_{\mathbb{M}\mathrm{an}})^{\mathsf{T}}$
minimal	deictic

Anaphora with World Knowledge

- **Example 2.10 (Anaphora with World Knowledge).**
  - *Mary is married to Jeff. Her husband is not in town.*
  - $\delta U_{\mathbb{F}}, V_{\mathbb{M}}.U = \mathrm{mary} \wedge \mathrm{married}(U, V) \wedge V = \mathrm{jeff}\ ;;$
    $\delta W_{\mathbb{M}}, W'_{\mathbb{F}}.\mathrm{husband}(W, W') \wedge \neg\mathrm{intown}(W)$
  - World knowledge
    - If a female X is married to a male Y, then Y is X's only husband.
    - $\rightsquigarrow \forall X_{\mathbb{F}}, Y_{\mathbb{M}}.\mathrm{married}(X, Y) \Rightarrow \mathrm{husband}(Y, X) \wedge (\forall Z.\mathrm{husband}(Z, X) \Rightarrow Z = Y)$
  - Model generation yields saturated tableau, all branches contain

$$\langle\{U, V, W, W'\}, \{[\mathrm{mary}/U], [\mathrm{jeff}/V], [\mathrm{jeff}/W], [\mathrm{mary}/W']\}, \mathcal{H}\rangle$$

with

$$\mathcal{H} = \{\mathrm{married}(\mathrm{mary}, \mathrm{jeff})^{\mathsf{T}}, \mathrm{husband}(\mathrm{jeff}, \mathrm{mary})^{\mathsf{T}}, \neg\mathrm{intown}(\mathrm{jeff})^{\mathsf{T}}\}$$

  - They only differ in additional negative facts, e.g. $\mathrm{married}(\mathrm{mary}, \mathrm{mary})^{\mathsf{F}}$.

Model Generation models Discourse Understanding

- The tableau machine algorithm conforms with psycholinguistic findings:
  - Zwaan& Radvansky [ZR98]: listeners not only represent logical form, but also models containing referents.
  - deVega [de 95]: online, incremental process.
  - Singer [Sin94]: enriched by background knowledge.
  - Glenberg et al. [GML87]: major function is to provide basis for anaphor resolution.

Chapter 10
Propositional Attitudes and Modalities

10.1 Introduction

Modalities and Propositional Attitudes

- **Definition 1.1.** Modality is a feature of language that allows for communicating things about, or based on, situations which need not be actual. A sentence is called modal, if it involves a modality
- **Definition 1.2.** Modality is signaled by phrases (called moods) that express a speaker's general intentions and commitment to how believable, obligatory, desirable, or actual an expressed proposition is.
- **Example 1.3.** Data on modalities (moods in red)
  - A *probably holds*, (possibilistic)
  - *it has always been the case that* A, (temporal)
  - *it is well-known that* A, (epistemic)
  - A *is allowed/prohibited*, (deontic)
  - A *is provable*, (provability)
  - A *holds after the program P terminates*, (program)
  - A *hods during the execution of P*. (dito)
  - *it is necessary that* A, (aletic)
  - *it is possible that* A, (dito)

Modeling Modalities and Propositional Attitudes

- **Example 1.4.** Again, the pattern from above:
  - *it is necessary that Peter knows logic* (A = Peter knows logic)
  - *it is possible that John loves logic*, (A = John loves logic)
- **Observation:** All of the red parts above modify the clause/sentence A. We call them modalities.
- **Definition 1.5 (A related Concept from Philosophy).** A propositional attitude is a mental state held by an agent toward a proposition.
- **Question:** But how to model this in logic?
- **Idea:** New sentence-to-sentence operators for *necessary* and *possible*. (extend existing logics with them.)
- **Observation:** A *is necessary*, iff $\neg$A *is impossible*.
- **Definition 1.6.** A modal logic is a logical system that has logical constants that model modalities.

History of Modal Logic

- Aristoteles studies the logic of necessity and possibility
- Diodorus: temporal modalities
  - possible: *is true or will be*
  - necessary: *is true and will never be false*
- Clarence Irving Lewis 1918 [Lew18] (Systems $S1, \ldots, S5$)
  - strict implication $I(\mathrm{A} \wedge \mathrm{B})$ (I for “impossible”)
- Kurt Gödel 1932: Modal logic of provability (S4) [Göd32]
- Saul Kripke 1959-63: Possible worlds semantics [Kri63]
- Vaugham Pratt 1976: Dynamic Program Logic [Pra76]
- $\vdots$

Basic Modal Logics (ML^0 and ML^1)

- **Definition 1.7.** Propositional modal logic ML^0 extends propositional logic with two new logical constants: $\Box$ for necessity and $\Diamond$ for possibility.($\Diamond A = \neg(\Box\neg A)$)
- **Observation:** Nothing hinges on the fact that we use propositional logic!
- **Definition 1.8.** First-order modal logic ML^1 extends first-order logic with two new logical constants: $\Box$ for necessity and $\Diamond$ for possibility.
- **Example 1.9.** We interpret
  1. *Necessarily, every mortal will die.* as $\Box(\forall X.\mathrm{mortal}(X) \Rightarrow \mathrm{willdie}(X))$
  2. *Possibly, something is immortal.* as $\Diamond(\exists X.\neg\mathrm{mortal}(X))$
- **Questions:** What do $\Box$ and $\Diamond$ mean? How do they behave?

Epistemic and Doxastic Modality

- **Definition 1.10.** Modal sentences can convey information about the speaker's state of knowledge (epistemic state) or belief (doxastic state).
- **Example 1.11.** We might paraphrase sentence (2) as (3):
  1. A: *Where's John?*
  2. B: *He might be in the library.*
  3. B': *It is consistent with the speaker's knowledge that John is in the library.*
- **Definition 1.12.** We way that a world w is an epistemic possibility for an agent B if it could be consistent with B's knowledge.
- **Definition 1.13.** An epistemic logic is one that models the epistemic state of a speaker. Doxastic logic does the same for the doxastic state.
- **Definition 1.14.** In deontic logic, we interpret the accessibility relation $\mathcal{R}$ as epistemic accessibility:
  - With this $\mathcal{R}$, represent B's utterance as $\Diamond \mathrm{inlib}(j)$.
  - Similarly, represent *John must be in the library.* as $\Box \mathrm{inlib}(j)$.
- **Question:** If $\mathcal{R}$ is epistemic accessibility, what properties should it have?

Deontic Modality

- **Definition 1.15.** Deontic modality is a modality that indicates how the world ought to be according to certain norms, expectations, speaker desire, etc.
- **Definition 1.16.** Deontic modality has the following subcategories
  - Commissive modality (the speaker's commitment to do something, like a promise or threat): e.g. *I shall help you*.
  - Directive modality (commands, requests, etc.): e.g. *Come!*, *Let's go!*, *You've got to taste this curry!*
  - Volitive modality (wishes, desires, etc.): *If only I were rich!*
- **Question:** If we want to interpret $\square \mathrm{runs}(j)$ as *It is required that John runs* (or, more idiomatically, as *John must run*), what formulae should be valid on this interpretation of the operators? (This is for homework!)

10.2 Semantics for Modal Logics

Semantics of ML^0

- **Definition 2.1.** We use a set $\mathcal{W}$ of possible worlds, and a accessibility relation $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$: if $\mathcal{R}(v, w)$, then we say that w is accessible from v.
- **Example 2.2.** $\mathcal{W} = \mathbb{N}$ with $\mathcal{R} = \{\langle n, n+1 \rangle \mid n \in \mathbb{N}\}$. (temporal logic)
- **Definition 2.3.** Variable assignment $\varphi \colon \mathcal{V}_0 \times \mathcal{W} \to \mathcal{D}_0$ assigns values to variables in a given possible world.
- **Definition 2.4.** Value function $\mathcal{I}_{\cdot}^{\cdot} \colon \mathcal{W} \times \mathit{wff}_0(\mathcal{V}_0) \to \mathcal{D}_0$ (assigns values to formulae in a possible world)
  - $\mathcal{I}_\varphi^w(V) = \varphi(w, V)$ for $.V \in \mathcal{V}_0$
  - $\mathcal{I}_\varphi^w(\neg \mathrm{A}) = \mathsf{T}$, iff $\mathcal{I}_\varphi^w(\mathrm{A}) = \mathsf{F}$. ($\wedge$ analogous)
  - $\mathcal{I}_\varphi^w(\Box \mathrm{A}) = \mathsf{T}$, iff $\mathcal{I}_\varphi^{w'}(\mathrm{A}) = \mathsf{T}$ for all $w' \in \mathcal{W}$ with $w \mathcal{R} w'$.
- **Definition 2.5.** We call a triple $\mathcal{M} := \langle \mathcal{W}, \mathcal{R}, \mathcal{I} \rangle$ a Kripke model.

Accessibility Relations. E.g. for Temporal Modalities

- **Example 2.6 (Temporal Worlds with Ordering).** Let $\langle\mathcal{W},\circ,<,\subseteq\rangle$ an interval time structure, then we can use $\langle\mathcal{W},<\rangle$ as a Kripke models. Then PAST becomes a modal operator.
- **Example 2.7.** Suppose we have $i<j$ and $j<k$. Then intuitively, if *Jane is laughing* is true at i, then *Jane laughed* should be true at j and at k, i.e. $\mathcal{I}_\varphi^w(j)\mathrm{PAST}(\mathrm{laughs}(j))$ and $\mathcal{I}_\varphi^w(k)\mathrm{PAST}(\mathrm{laughs}(j))$.
  But this holds only if "$<$" is transitive. (which it is!)
- **Example 2.8.** Here is a clearly counter-intuitive claim: For any time i and any sentence A, if $\mathcal{I}_\varphi^w(i)\mathrm{PRES}(\mathsf{A})$ then $\mathcal{I}_\varphi^w(i)\mathrm{PAST}(\mathsf{A})$.
  (For example, the truth of *Jane is at the finish line* at i implies the truth of *Jane was at the finish line* at i.)
  But we would get this result if we allowed $<$ to be reflexive. ($<$ is irreflexive)
- Treating tense modally, we obtain reasonable truth conditions.

Modal Axioms (Propositional Logic)

- **Definition 2.9.** Necessitation: $\frac{A}{\Box A}\,N$
- **Definition 2.10 (Normal Modal Logics).**

System	Axioms	Accessibility Relation
$\mathbb{K}$	$\Box(A \Rightarrow B) \Rightarrow (\Box A \Rightarrow \Box B)$	general
$\mathbb{T}$	$\mathbb{K} + \Box A \Rightarrow A$	reflexive
$\mathbb{S}4$	$\mathbb{T} + \Box A \Rightarrow \Box\Box A$	reflexive + transitive
$\mathbb{B}$	$\mathbb{T} + \Diamond\Box A \Rightarrow A$	reflexive + symmetric
$\mathbb{S}5$	$\mathbb{S}4 + \Diamond A \Rightarrow \Box\Diamond A$	equivalence relation

$\mathbb{K}$ Theorems

- **Observation 2.11.** $\Box(A \wedge B) \models \Box A \wedge \Box B$ *in* $\mathbb{K}$*.*
- **Observation 2.12.** $A \Rightarrow B \models \Box A \Rightarrow \Box B$ *in* $\mathbb{K}$*.*
- **Observation 2.13.** $A \Rightarrow B \models \Diamond A \Rightarrow \Diamond B$ *in* $\mathbb{K}$*.*

Translation to First-Order Logic

- **Question:** Is modal logic more expressive than predicate logic?
- **Answer:** Very rarely! (usually can be translated)
- **Definition 2.14.** Translation τ from ML into PL^1, (so that the diagram commutes)

$$\begin{tikzcd}
\text{Kripke-Sem.} \arrow[r, "\overline{\tau}"] & \text{Tarski-Sem.} \\
\text{modal logic} \arrow[u, "\mathcal{I}_{\varphi}^{w}"] \arrow[r, "\tau"] & \text{predicate logic} \arrow[u, "\mathcal{I}_{\varphi}"']
\end{tikzcd}$$

- **Idea:** Axiomatize Kripke models in PL^1. (diagram is simple consequence)
- **Definition 2.15.** A logic morphism $\Theta\colon \mathcal{L} \to \mathcal{L}'$ is called
  - correct, iff $\exists \mathcal{M}.\mathcal{M} \models \Phi$ implies $\exists \mathcal{M}'.\mathcal{M}' \models' \Theta(\Phi)$.
  - complete, iff $\exists \mathcal{M}'.\mathcal{M}' \models' \Theta(\Phi)$ implies $\exists \mathcal{M}.\mathcal{M} \models \Phi$.

Modal Logic Translation (formal)

- **Definition 2.16.** The standard translation τ_w from modal logics to first-order logic is given by the following process:
  - Extend all function constants by a "world argument": $\overline{f} \in \Sigma^f_{k+1}$ for every $f \in \Sigma^f_k$
  - for predicate constants accordingly.
  - insert the "translation world" there: e.g. $\tau_w(f(a,b)) = \overline{f}(w, \overline{a}(w), \overline{b}(w))$.
  - New predicate constant $\mathcal{R}$ for the accessibility relation.
  - New constant s for the "start world".
  - $\tau_w(\Box \mathrm{A}) = \forall w'.w\mathcal{R}w' \Rightarrow \tau_{w'}(\mathrm{A})$.
  - Use all axioms from the respective correspondence theory.
- **Definition 2.17 (Alternative).** Functional translations, if $\mathcal{R}$ associative:
  - New function constant $f_{\mathcal{R}}$ for the accessibility relation.
  - Revise the standard translation by one of the following
    - $\tau_w(\Box \mathrm{A}) = \forall w'.w = f_{\mathcal{R}}(w') \Rightarrow \tau_w(\mathrm{A})$. (naive solution)
    - $\tau_{f_{\mathcal{R}}(w)}(\Box \mathrm{A}) = \tau_w(\mathrm{A})$ (better for mechanizing [Ohl88])

Translation (continued)

- **Theorem 2.18.** $\tau_s\colon \mathrm{ML}^0 \to \mathrm{PL}^0$ *is correct and complete.*
- *Proof:* show that $\exists\mathcal{M}.\mathcal{M} \models \Phi$ iff $\exists\mathcal{M}'.\mathcal{M}' \models \tau_s(\Phi)$
  1. Let $\mathcal{M} = \langle\mathcal{W},\mathcal{R},\varphi\rangle$ with $\mathcal{M} \models \mathsf{A}$
  2. chose $\mathcal{M} = \langle\mathcal{W},\mathcal{I}'\rangle$, such that $\mathcal{I}(\overline{p}) = \varphi(p)\colon \mathcal{W} \to \{\mathsf{T},\mathsf{F}\}$ and $\mathcal{I}(r) = \mathcal{R}$.

  we prove $\mathcal{M} \models_\psi \tau_w(\mathsf{A})'$ *for* $\psi = \mathrm{Id}_\mathcal{W}$ *by structural induction over* A*.*

  3. $\mathsf{A} = P$
     - 3.1. $\mathcal{I}_\psi(\tau_w(\mathsf{A})) = \mathcal{I}_\psi(\overline{p}(w)) = \mathcal{I}(\overline{p}(w)) = \varphi(P,w) = \mathsf{T}$
  4. $\mathsf{A} = \neg\mathsf{B}$, $\mathsf{A} = \mathsf{B}\wedge\mathsf{C}$ *trivial by IH.*
  5. $\mathsf{A} = \Box\mathsf{B}$
     - 5.1. $\mathcal{I}_\psi(\tau_w(\mathsf{A})) = \mathcal{I}_\psi(\forall w.r(w,v) \Rightarrow \tau_v(\mathsf{B})) = \mathsf{T}$, if $\mathcal{I}_\psi(r(w,v)) = \mathsf{F}$ or $\mathcal{I}_\psi(\tau_v(\mathsf{B})) = \mathsf{T}$ for all $v \in \mathcal{W}$.
     - 5.2. $\mathcal{M} \models_\psi \tau_{v'}(\mathsf{B})$ so by IH $\mathcal{M} \models^v \mathsf{B}$.
     - 5.3. so $\mathcal{M} \models_\psi \tau_w(\mathsf{A})'$.

Modal Logic (References)

- G. E. Hughes und M. M. Cresswell: *A companion to Modal Logic*, University Paperbacks, Methuen (1984) [HC84].
- David Harel: *Dynamic Logic*, Handbook of Philosophical Logic, D. Gabbay, Hrsg. Reidel (1984) [Har84].
- Johan van Benthem: *Language in Action, Categories, Lambdas and Dynamic Logic*, North Holland (1991) [Ben91].
- Reinhard Muskens, Johan van Benthem, Albert Visser, *Dynamics*, in *Handbook of Logic and Language*, Elsevier, (1995) [MBV95].
- Blackburn, DeRijke, Vedema: *Modal Logic*; 2001 [BRV01]. look at the chapter "Guide to the literature" in the end.

10.3 A Multiplicity of Modalities $\leadsto$ Multimodal Logic

A Multiplicity of Modalities

- Epistemic (knowledge and belief) modalities must be relativized to an individual
  - *Peter knows that Trump is lying habitually.*
  - *John believes that Peter knows that Trump is lying habitually.*
  - *You must take the written drivers' exam to be admitted to the practical test.*
- Similarly, we find in natural language expressions of necessity and possibility relative to many different kinds of things.
- Consider the deontic (obligatory/permissible) modalities
  - *[Given the university's rules] Jane can take that class.*
  - *[Given her intellectual ability] Jane can take that class.*
  - *[Given her schedule] Jane can take that class.*
  - *[Given my desires] I must meet Henry.*
  - *[Given the requirements of our plan] I must meet Henry.*
  - *[Given the way things are] I must meet Henry [every day and not know it].*
- Many different sorts of modality, sentences are multiply ambiguous towards which one.

Multimodal Logics

- **Definition 3.1.** A multimodal logic provides operators for multiple modalities: $[1], [2], [3], \ldots, \langle 1\rangle, \langle 2\rangle, \ldots$
- **Definition 3.2.** Multimodal Kripke models provide multiple accessibility relations $\mathcal{R}_1, \mathcal{R}_2, \ldots \subseteq \mathcal{W} \times \mathcal{W}$.
- **Definition 3.3.** The value function in multimodal logic generalizes the clause for $\Box$ in ML^0 to
  - $\mathcal{I}_\varphi^w([i]\mathrm{A}) = \mathsf{T}$, iff $\mathcal{I}_\varphi^{w'}(\mathrm{A}) = \mathsf{T}$ for all $w' \in \mathcal{W}$ with $w\mathcal{R}_i w'$.
- **Example 3.4 (Epistemic Logic: talking about knowing/believing).** $[peter]\langle klaus\rangle\mathrm{A}$ (Peter knows that Klaus considers A possible)
- **Example 3.5 (Program Logic: talking about programs).** $[X{:=}\mathrm{A}][Y{:=}\mathrm{A}]X = Y$ (after assignments, the values of X and Y are equal)

10.4 Dynamic Logic for Imperative Programs

Dynamic Program Logic (DL)

- Modal logics for argumentation about imperative, non-deterministic programs.
- **Idea:** Formalize the traditional argumentation about program correctness: tracing the variable assignments (state) across program statements.
- **Example 4.1 (Fibonacci).** Consider the following (imperative) program that computes $\mathrm{Fib}(X)$ as the value of Z:
  $\alpha := \langle Y, Z\rangle{:=}\langle 1, 1\rangle \mathrel{;} \text{while } X \neq 0 \text{ do } \langle X, Y, Z\rangle{:=}\langle X-1, Z, Y+Z\rangle \text{ end}$
  - States for the "input" $X = 4$: $\langle 4, \_, \_\rangle, \langle 4, 1, 1\rangle, \langle 3, 1, 2\rangle, \langle 2, 2, 3\rangle, \langle 1, 3, 5\rangle, \langle 0, 5, 8\rangle$
  - **Correctness**? For positive X, running α with input $\langle X, \_, \_\rangle$ we end with $\langle 0, \mathrm{Fib}(X-1), \mathrm{Fib}(X)\rangle$
  - Termination? α does not terminate on input $\langle -1, \_, \_\rangle$.

Multi-Modal Logic fits well

- **Observation:** Multi modal logic fits well
  - States as possible worlds, program statements as accessibility relations.
  - Two syntactic categories: programs α and formulae A.
  - Interpret $[\alpha]$A as *If α terminates, then A holds afterwards*
  - Interpret $\langle\alpha\rangle$A as *α terminates and A holds afterwards*.
- **Example 4.2.** Assertions about Fibonacci number (α)
  - $\forall X, Y.[\alpha]Z = \mathrm{Fib}(X)$
  - $\forall X, Y.(X \geq 0) \Rightarrow \langle\alpha\rangle Z = \mathrm{Fib}(X)$

Levels of Description in Dynamic Program Logic

- Propositional dynamic logic (DL^0) (independent of variable assignments)
  - $\models [\alpha]\mathsf{A} \wedge [\alpha]\mathsf{B} \Leftrightarrow [\alpha](\mathsf{A} \wedge \mathsf{B})$
  - $\models [\text{while } \mathsf{A} \vee \mathsf{B} \text{ do } \alpha \text{ end}]\mathsf{C} \Leftrightarrow [\text{while } \mathsf{A} \text{ do } \alpha \text{ end} \,;\, \text{while } \mathsf{B} \text{ do } \alpha \,;\, \text{while } \mathsf{A} \text{ do } \alpha \text{ end end}]\mathsf{C}$
- First-order program logic (DL^1) (function, predicates uninterpreted)
  - $\models p(f(X)) \Rightarrow g(Y, f(X)) \Rightarrow \langle Z{:=}f(X)\rangle p(Z, g(Y, Z))$
  - $\models Z = Y \wedge (\forall X. f(g(X)) = X) \Rightarrow [\text{while } p(Y) \text{ do } Y{:=}g(Y) \text{ end}]\langle \text{while } Y \neq Z \text{ do } Y{:=}f(Y) \text{ end}\rangle T$
- DL^1 with interpreted functions, predicates (maybe some other time)
  - $\forall X.\langle \text{while } X \neq 1 \text{ do if } even(X) \text{ then} X{:=}\frac{X}{2} \text{ else } X{:=}3X + 1 \text{ end}\rangle T$
- **Definition 4.3.** We collectively call these dynamic program logics.

DL^0 Syntax

- **Definition 4.4.** Propositional dynamic logic (DL^0) is PL^0 extended by
  - program variables $\mathcal{V}_\pi = \{\alpha, \beta, \gamma, \ldots\}$,
  - modalities $[\alpha]$, $\langle\alpha\rangle$.
  - program constructors $\Sigma^\pi = \{;, \cup, *, ?\}$ (minimal set)

$\alpha\,;\,\beta$	execute first α, then β	sequence
$\alpha \cup \beta$	execute (non-deterministically) either α or β	distribution
$*\alpha$	(non-deterministically) repeat α finitely often	iteration
A?	proceed if $\models$ A, else stop	test

- **Idea:** Standard program primitives as derived concepts

Construct	as
if A thenα else β	$(\mathrm{A}?\,;\,\alpha) \cup (\neg\mathrm{A}?\,;\,\beta)$
while A do α end	$*(\mathrm{A}?\,;\,\alpha)\,;\,\neg\mathrm{A}?$
repeat α until A end	$*(\alpha\,;\,\neg\mathrm{A}?)\,;\,\mathrm{A}?$

DL^0 Semantics

- **Definition 4.5.** A model for DL^0 consists of a set $\mathcal{W}$ of possible worlds called states for DL^0.
- **Definition 4.6.** DL^0 variable assignments come in two parts:
  - $\varphi \colon \mathcal{V}_0 \times \mathcal{W} \to \mathcal{D}_0$ (for propositional variables)
  - $\pi \colon \mathcal{V}_\pi \to \mathcal{P}(\mathcal{W} \times \mathcal{W})$ (maps program variables to accessibility relations)
- **Definition 4.7.** The meaning of complex formulae is given by the following value function $\mathcal{I}^w_{\varphi,\pi} \colon wf\!f_0(\mathcal{V}_0) \to \mathcal{D}_0$ on formulae:
  - $\mathcal{I}^w_{\varphi,\pi}(V) = \varphi(w, V)$ for $V \in \mathcal{V}_0$.
  - $\mathcal{I}^w_{\varphi,\pi}(\neg \mathrm{A}) = \mathsf{T}$ iff $\mathcal{I}^w_{\varphi,\pi}(\mathrm{A}) = \mathsf{F}$
  - $\mathcal{I}^w_{\varphi,\pi}([\alpha]\mathrm{A}) = \mathsf{T}$ iff $\mathcal{I}^{w'}_{\varphi,\pi}(\mathrm{A}) = \mathsf{T}$ for all $w' \in \mathcal{W}$ with $w\mathcal{I}_{\varphi,\pi}(\alpha)w'$.

  And $\mathcal{I}_{\varphi,\pi} \colon wf\!f_0(\mathcal{V}_0) \to \mathcal{P}(\mathcal{W} \times \mathcal{W})$ on programs: (independent of $w \in \mathcal{W}$)
  - $\mathcal{I}_{\varphi,\pi}(\alpha) = \pi(\alpha)$. (program variable by assignment)
  - $\mathcal{I}_{\varphi,\pi}(\alpha\,;\beta) = \mathcal{I}_{\varphi,\pi}(\beta) \circ \mathcal{I}_{\varphi,\pi}(\alpha)$ (sequence by composition)
  - $\mathcal{I}_{\varphi,\pi}(\alpha \cup \beta) = \mathcal{I}_{\varphi,\pi}(\alpha) \cup \mathcal{I}_{\varphi,\pi}(\beta)$ (distribution by union)
  - $\mathcal{I}_{\varphi,\pi}(*\alpha) = \mathcal{I}_{\varphi,\pi}(\alpha)^*$ (iteration by reflexive transitive closure)
  - $\mathcal{I}_{\varphi,\pi}(\mathrm{A}?) = \{\langle w, w\rangle \mid \mathcal{I}^w_{\varphi,\pi}(\mathrm{A}) = \mathsf{T}\}$ (test by subset of identity relation)

First-Order Program Logic (DL^1)

- **Observation:** Imperative programs uses variables, function and predicate constants (uninterpreted), but no program variables. The main operation is variable assignment.
- **Idea:** Make a multimodal logic in the spirit of DL^0 that features all of these for a deeper understanding.
- **Definition 4.8.** First-order program logic (DL^1) combines the features of PL^1, DL^0 without program variables, with the following two assignment operators:
  - nondeterministic assignment $X{:=}?$
  - deterministic assignment $X{:=}\mathrm{A}$
- **Example 4.9.** $\models p(f(X)) \Rightarrow g(Y, f(X)) \Rightarrow \langle Z{:=}f(X)\rangle p(Z, g(Y, Z))$ in DL^1.
- **Example 4.10.** In DL^1 we have
  $\models Z = Y \wedge (\forall X.p(f(g(X)) = X)) \Rightarrow [\text{while } p(Y) \text{ do } Y{:=}g(Y) \text{ end}]\langle \text{while } Y \neq Z \text{ do } Y{:=}f(Y) \text{ end}\rangle T$

DL^1 Semantics

- **Definition 4.11.** Let $\mathcal{M} = \langle\mathcal{D},\mathcal{I}\rangle$ be a first-order model then the states (possible worlds) are variable assignments: $\mathcal{W} = \{\varphi \,|\, \varphi\colon \mathcal{V}_\iota \to \mathcal{D}\}$
- **Definition 4.12.** For a set $\mathcal{X}$ of variables, write $\varphi[\mathcal{X}]\psi$, iff $\varphi(X) = \psi(X)$ for all $X \notin \mathcal{X}$.
- **Definition 4.13.** The meaning of complex formulae is given by the following value function $\mathcal{I}_\varphi^w\colon \mathit{wff}_o(\Sigma,\mathcal{V}_\iota) \to \mathcal{D}_0$
  - $\mathcal{I}_\varphi^w(\mathrm{A}) = \mathcal{I}_\varphi(\mathrm{A})$ if A term or atom.
  - $\mathcal{I}_\varphi^w(\neg\mathrm{A}) = \mathsf{T}$ iff $\mathcal{I}_\varphi^w(\mathrm{A}) = \mathsf{F}$
  - $\vdots$
  - $\mathcal{I}_\varphi(X{:=}?) = \{\langle\varphi,\psi\rangle \,|\, \varphi[X]\psi\}$
  - $\mathcal{I}_\varphi(X{:=}\mathrm{A}) = \{\langle\varphi,\psi\rangle \,|\, \varphi[X]\psi \text{ and } \psi(X) = \mathcal{I}_\varphi(\mathrm{A})\}$.
- **Observation 4.14 (Substitution and Quantification).** *We have*
  - $\mathcal{I}_\varphi([X{:=}\mathrm{A}]\mathrm{B}) = \mathcal{I}_{\varphi,[\mathcal{I}_\varphi(\mathrm{A})/X]}(\mathrm{B})$
  - $\forall X.\mathrm{A} = [X{:=}?]\mathrm{A}$.
- Thus substitutions and quantification are definable in DL^1.

Natural Deduction for DL^0

- **Definition 4.15.** The natural deduction calculus $\mathcal{DND}_0$ for DL^0 contains the inference rules from $\mathcal{ND}_0$ plus:

Introduction

Elimination

$$\frac{[\alpha][\beta]\mathrm{A}}{[\alpha\,;\,\beta]\mathrm{A}}\;\mathcal{DND}_0;I \qquad\qquad \frac{[\alpha\,;\,\beta]\mathrm{A}}{[\alpha][\beta]\mathrm{A}}\;\mathcal{DND}_0;E$$

$$\frac{[\alpha]\mathrm{A}\quad[\beta]\mathrm{A}}{[\alpha\cup\beta]\mathrm{A}}\;\mathcal{DND}_0\cup I \qquad \frac{[\alpha\cup\beta]\mathrm{A}}{[\alpha]\mathrm{A}}\;\mathcal{DND}_0\cup E_l \quad \frac{[\alpha\cup\beta]\mathrm{A}}{[\beta]\mathrm{A}}\;\mathcal{DND}_0\cup E_r$$

$$\frac{[\alpha]^0\mathrm{A}\quad\ldots\quad[\alpha]^n\mathrm{A}\;\text{ for all } n\in\mathbb{N}}{[*\alpha]\mathrm{A}}\;\mathcal{DND}_0*I \qquad \frac{[*\alpha]\mathrm{A}\quad n\in\mathbb{N}}{[\alpha]^n\mathrm{A}}\;\mathcal{DND}_0*E$$

$$\frac{\begin{array}{c}[\mathrm{A}]^1\\ \hline\hline \mathrm{B}\end{array}}{[\mathrm{A}?]\mathrm{B}}\;\mathcal{DND}_0?I^1 \qquad\qquad \frac{[\mathrm{A}?]\mathrm{B}\quad\mathrm{A}}{\mathrm{B}}\;\mathcal{DND}_0?E$$

For details see [HM95].

Natural Deduction for DL^1

- **Definition 4.16.** The natural deduction calculus $\mathcal{DND}_1$ for DL^1 contains the inference rules from $\mathcal{ND}^1$ and $\mathcal{DND}_0$ plus:

$$\frac{[\mathsf{A}/X](\mathsf{B}) \quad X \notin (\mathrm{free}(\mathsf{A}) \cup \mathrm{free}(\mathsf{B}))}{[X{:=}\mathsf{A}]\mathsf{B}} \; \mathcal{DND}_0{:=}I$$

$$\frac{[X{:=}\mathsf{A}]\mathsf{B} \qquad \begin{array}{c}[[\mathsf{A}/X](\mathsf{B})]^1 \\ \hline\hline \mathsf{C}\end{array}}{\mathsf{C}} \; \mathcal{DND}_0{:=}E$$

  For details see [HM95].

- **Observation:** No inference rules for $:=?$ needed as $\forall X.\mathsf{A} = [X{:=}?]\mathsf{A}$ $\leadsto$ $\mathcal{ND}^1\forall I$ and $\mathcal{ND}^1\forall E$ suffice.

Natural Language as Programming Languages

- **Question:** Why is dynamic program logic interesting in a natural language semantics course?
- **Answer:** There are fundamental relations between dynamic (discourse) logics and dynamic program logics.
- **David Israel:** "*Natural languages are programming languages for mind*" [Isr93]

Chapter 11
Some Issues in the Semantics of Tense

Tense as a Deictic Element

- **Goal:** Capturing the truth conditions and the logical form of sentences of English.
- **Clearly:** The following three sentences have different truth conditions.
  1. *Jane saw George.*
  2. *Jane sees George.*
  3. *Jane will see George.*
- **Observation 0.1.** *Tense is a deictic element, i.e. its interpretation requires reference to something outside the sentence itself.*
- **Remark:** Often, in particular in the case of monoclausal sentences occurring in isolation, as in our examples, this "something" is the speech time.
- **Idea:** make use of the reference time *now*:
  - *Jane saw George* is true at a time iff *Jane sees George* was true at some point in time before now.
  - *Jane will see George* is true at a time iff *Jane sees George* will be true at some point in time after now.

A Simple Semantics for Tense

- **Problem:** The meaning of *Jane saw George* and *Jane will see George* is defined in terms of *Jane sees George*.
  ⤳ We need the truth conditions of the present tense sentence.
- **Idea:** *Jane sees George* is true at a time iff Jane sees George at that time.
- **Implementation:** Postulate temporal operator as sentential operators (expressions of type $o \to o$). Interpret
  1. *Jane saw George* as $\mathrm{PAST}(\mathrm{see}(g,j))$,
  2. *Jane sees George* as $\mathrm{PRES}(\mathrm{see}(g,j))$, and
  3. *Jane wil see George* as $\mathrm{FUT}(\mathrm{see}(g,j))$.

Models and Evaluation for a Tensed Language

- **Problem:** The interpretations of constants vary over time.
- **Idea:** Introduce times into our models, and let the interpretation function give values of constants at a time. Relativize the valuation function to times
- **Idea:** We will consider temporal structures, where denotations are constant on intervals.
- **Definition 0.2.** Let $I \subseteq \{[i,j] \,|\, i,j \in \mathbb{R}\}$ be a set of real intervals, then we call $\langle I, \circ, <, \subseteq \rangle$ an interval time structure, where for intervals $i := [i_l,i_l]$ and $j := [l_l,j_r]$ we say that
  - i and j overlap (written $i \circ j$), iff $l_l \leq ir$,
  - i precedes j (written $i < j$), iff $ir \leq l_l$, and
  - i is contained in j (written $i \subseteq j$), iff $l_l \leq i_l$ and $ir \leq j_r$.
- **Definition 0.3.** A temporal model is a triple $\langle \mathcal{D}, \mathbb{I}, \mathcal{I} \rangle$, where
  - $\mathcal{D}$ is a set called the domain,
  - $\mathbb{I}$ is an interval time structure, and
  - $\mathcal{I} \colon \mathbb{I} \times \Sigma_{\mathcal{T}} \to \mathcal{D}$ an interpretation function.

Interpretation rules for the temporal operators

- **Definition 0.4.** For the value function $\mathcal{I}_\varphi^i(\cdot)$ we only redefine the clause for constants:
  - $\mathcal{I}_\varphi^i(c) := \mathcal{I}^i(c)$
  - $\mathcal{I}_\varphi^i(X) := \varphi(X)$
  - $\mathcal{I}_\varphi^i(\mathsf{FA}) := \mathcal{I}_\varphi^i(\mathsf{F})(\mathcal{I}_\varphi^i(\mathsf{A}))$.
- **Definition 0.5.** We define the meaning of the temporal operators:
  1. $\mathcal{I}_\varphi^i(\mathrm{PRES}(\Phi)) = \mathsf{T}$, iff $\mathcal{I}_\varphi^i(\Phi) = \mathsf{T}$.
  2. $\mathcal{I}_\varphi^i(\mathrm{PAST}(\Phi)) = \mathsf{T}$ iff there is an interval $j \in \mathbb{I}$ with $j < i$ and $\mathcal{I}_\varphi^j(\Phi) = \mathsf{T}$.
  3. $\mathcal{I}_\varphi^i(\mathrm{FUT}(\Phi)) = \mathsf{T}$ iff there is an interval $j \in \mathbb{I}$ with $i < j$ and $\mathcal{I}_\varphi^j(\Phi) = \mathsf{T}$.

Complex Tenses in English

- How do we use this machinery to deal with complex tenses in English?
  - Past of past (pluperfect): *Jane had left (by the time I arrived)*.
  - Future perfect: *Jane will have left (by the time I arrive)*.
  - Past progressive: *Jane was going to leave (when I arrived)*.

Perfective vs. imperfective

- **Data:**
  - *Jane left.*
  - *Jane was leaving.*
- **Question:** How do the truth conditions of these sentences differ?
- **Standard observation:**
  - Perfective indicates a completed action,
  - imperfective indicates an incomplete or ongoing action.
- This becomes clearer when we look at the “creation predicates” like *build a house* or *write a book*
  - *Jane built a house.* entails: *There was a house that Jane built.*
  - *Jane was building a house.* does not entail that *there was a house that Jane built.*

Future Readings of Present Tense

- **New Data:**
  1. *Jane leaves tomorrow.*
  2. *Jane is leaving tomorrow.*
  3. ?? *It rains tomorrow.*
  4. ?? *It is raining tomorrow.*
  5. ?? *The dog barks tomorrow.*
  6. ??*The dog is barking tomorrow.*
- Future readings of present tense appear to arise only when the event described is planned, or planable, either by the subject of the sentence, the speaker, or a third party.

Sequence of Tense

- *George said that Jane was laughing.*
  - Reading 1: George said "*Jane is laughing.*" I.e. saying and laughing co-occur. So past tense in subordinate clause is past of utterance time, but not of main clause reference time.
  - Reading 2: George said "*Jane was laughing.*" I.e. laughing precedes saying. So past tense in subordinate clause is past of utterance time and of main clause reference time.

Sequence of Tense

- *George said that Jane was laughing.*
  - Reading 1: George said "*Jane is laughing.*" I.e. saying and laughing co-occur. So past tense in subordinate clause is past of utterance time, but not of main clause reference time.
  - Reading 2: George said "*Jane was laughing.*" I.e. laughing precedes saying. So past tense in subordinate clause is past of utterance time and of main clause reference time.
- *George saw the woman who was laughing.*
  - How many readings?

Sequence of Tense

- *George said that Jane was laughing.*
  - Reading 1: George said "*Jane is laughing.*" I.e. saying and laughing co-occur. So past tense in subordinate clause is past of utterance time, but not of main clause reference time.
  - Reading 2: George said "*Jane was laughing.*" I.e. laughing precedes saying. So past tense in subordinate clause is past of utterance time and of main clause reference time.
- *George saw the woman who was laughing.*
  - How many readings?
- *George will say that Jane is laughing.*
  - Reading 1: George will say "*Jane is laughing.*" Saying and laughing co-occur, but both saying and laughing are future of utterance time. So present tense in subordinate clause indicates futurity relative to utterance time, but not to main clause reference time.
  - Reading 2: Laughing overlaps utterance time and saying (by George). So present tense in subordinate clause is present relative to utterance time *and* main clause reference time.

Sequence of Tense (continued)

- *George will see the woman who is laughing.*
  - How many readings?
- Note that in all of the above cases, the predicate in the subordinate clause describes an event that is extensive in time. Consider readings when subordinate event is punctual.

Sequence of Tense (continued)

- *George will see the woman who is laughing.*
  - How many readings?
- Note that in all of the above cases, the predicate in the subordinate clause describes an event that is extensive in time. Consider readings when subordinate event is punctual.
- *George said that Mary fell.*
  - Falling must precede George's saying.

Sequence of Tense (continued)

- *George will see the woman who is laughing.*
  - How many readings?
- Note that in all of the above cases, the predicate in the subordinate clause describes an event that is extensive in time. Consider readings when subordinate event is punctual.
- *George said that Mary fell.*
  - Falling must precede George's saying.
- *George saw the woman who fell.*
  - Same three readings as before: falling must be past of utterance time, but could be past, present or future relative to seeing (i.e main clause reference time).

Sequence of Tense (continued)

- *George will see the woman who is laughing.*
  - How many readings?
- Note that in all of the above cases, the predicate in the subordinate clause describes an event that is extensive in time. Consider readings when subordinate event is punctual.
- *George said that Mary fell.*
  - Falling must precede George's saying.
- *George saw the woman who fell.*
  - Same three readings as before: falling must be past of utterance time, but could be past, present or future relative to seeing (i.e main clause reference time).
- And just for fun, consider past under present... *George will claim that Mary hit Bill.*
  - Reading 1: hitting is past of utterance time (therefore past of main clause reference time).
  - Reading 2: hitting is future of utterance time, but past of main clause reference time.

Sequence of Tense (continued)

- *George will see the woman who is laughing.*
  - How many readings?
- Note that in all of the above cases, the predicate in the subordinate clause describes an event that is extensive in time. Consider readings when subordinate event is punctual.
- *George said that Mary fell.*
  - Falling must precede George's saying.
- *George saw the woman who fell.*
  - Same three readings as before: falling must be past of utterance time, but could be past, present or future relative to seeing (i.e main clause reference time).
- And just for fun, consider past under present... *George will claim that Mary hit Bill.*
  - Reading 1: hitting is past of utterance time (therefore past of main clause reference time).
  - Reading 2: hitting is future of utterance time, but past of main clause reference time.
- And finally...
  1. *A week ago, John decided that in ten days at breakfast he would tell his mother that they were having their last meal together.* (Abusch 1988)
  2. *John said a week ago that in ten days he would buy a fish that was still alive.* (Ogihara 1996)

Interpreting Tense in Discourse

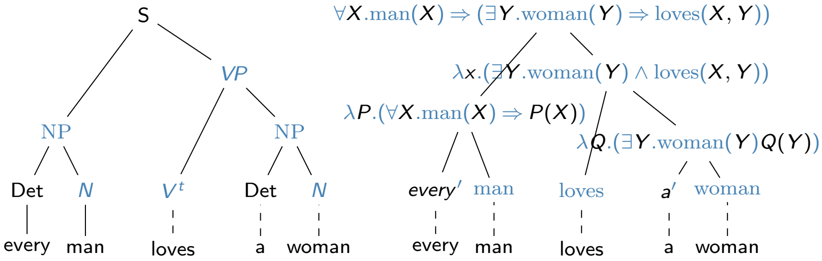
- **Example 0.6 (Ordering and Overlap).** *A man walked into the bar. He sat down and ordered a beer. He was wearing a nice jacket and expensive shoes, but he asked me if I could spare a buck.*
- **Example 0.7 (Tense as anaphora?).**
  1. Said while driving down the NJ turnpike: *I forgot to turn off the stove.*
  2. *I didn't turn off the stove.*

Chapter 12
Quantifier Scope Ambiguity and Underspecification

12.1 Scope Ambiguity and Quantifying-In

Quantifier Scope Ambiguities: Data

- Consider the following sentences:
  1. *Every man loves a woman* (Britney Spears or his mother?)
  2. *Most Europeans speak two languages.*
  3. *Some student in every course sleeps in every class at least some of the time.*
- **Definition 1.1.** We call these systematic ambiguities quantifyer scope ambiguities
- **Example 1.2.** We can represent the “wide-scope” reading with our methods



- **Question:** How to map an unambiguous input structure to multiple translations.

Storing and Quantifying In

- **Analysis:** The sentence meaning is of the form $\langle \text{everyman} \rangle(\langle \text{awoman} \rangle(\langle \text{loves} \rangle))$
- **Idea:** Somehow have to move the *a woman* part in front of the *every* to obtain

$$\langle \text{awoman} \rangle(\langle \text{everyman} \rangle(\langle \text{loves} \rangle))$$

- **More concretely:** Let's try *A woman - every man loves her.*
  In semantics construction, apply *a woman* to *every man loves her*.
  So *a woman* out-scopes *every man*.
- **Problem:** How to represent pronouns and link them to their antecedents
- STORE is an alternative translation rule. Given a node with an NP daughter, we can translate the node by passing up to it the translation of its non-NP daughter, and putting the translation of the NP into a store, for later use.
- The QI rule allows us to empty out a non-empty store.

Storing and Quantifying In (Technically)

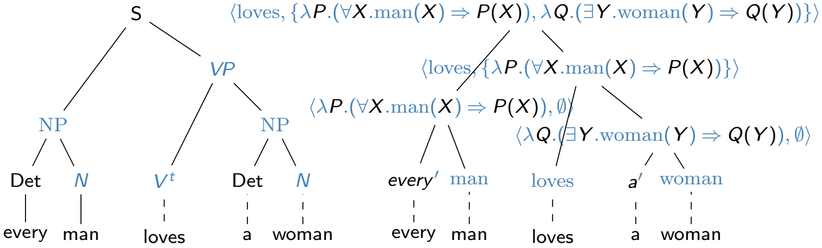
- **Definition 1.3.** $\mathrm{STORE}(NP,\Phi) \longrightarrow (\Phi,\Sigma * NP)$, where $\Sigma * NP$ is the result of adding NP to Σ, i.e. $\Sigma * NP = \Sigma \cup \{NP\}$; we will assume that NP is not already in Σ, when we use the $*$ operator.
- **Definition 1.4.** $\mathrm{QI}(\langle\Phi,\Sigma * NP\rangle) \rightarrow \langle NP \oplus \Phi,\Sigma\rangle$ where $\oplus$ is either function application or function composition.
- **Nondeterministic Semantics Construction:** Adding rules gives us more choice
  1. **Rule C (simple combination)** If A is a node with daughters B and C, and the translations of B and of C have empty stores, then A translates to $B' \oplus C'$. Choice of rule is determined by types.
  2. STORE If A is a node with daughters B and C, where:
     - B is an NP with translation B' and
     - C translates to (C',Σ)

     then A may translate to $\mathrm{STORE}(B',C')$

  Note that STORE may be applied whether or not the stores of the constituent nodes are empty.

Quantifying in Practice: *Every man loves a woman*

- **Example 1.5.**



- Continue with **QI** applications: first retrieve $\lambda Q.(\exists Y.\text{woman}(Y) \Rightarrow Q(Y))$

$$\begin{array}{ll}
 & \langle \text{loves}, \{\lambda P.(\forall X.\text{man}(X) \Rightarrow P(X)), \lambda Q.(\exists Y.\text{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{QI} & \langle \circ\ (\lambda P.(\forall X.\text{man}(X) \Rightarrow P(X)))\ \text{loves}, \{\lambda Q.(\exists Y.\text{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{\beta} & \langle \lambda Z.(\lambda P.(\forall X.\text{man}(X) \Rightarrow P(X)))\ \text{loves}\ Z, \{\lambda Q.(\exists Y.\text{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{\beta} & \langle \lambda Z.(\forall X.\text{man}(X) \Rightarrow \text{loves}\ Z\ X), \{\lambda Q.(\exists Y.\text{woman}(Y) \Rightarrow Q(Y))\}\rangle \\
\rightarrow_{QI} & \langle (\lambda Q.(\exists Y.\text{woman}(Y) \Rightarrow Q(Y)))\ (\lambda Z.(\forall X.\text{man}(X) \Rightarrow \text{loves}\ Z\ X)), \emptyset\rangle \\
\rightarrow_{\beta} & \langle \exists Y.\text{woman}(Y) \Rightarrow (\lambda Z.(\forall X.\text{man}(X) \Rightarrow \text{loves}\ Z\ X))\ Y, \emptyset\rangle \\
\rightarrow_{\beta} & \langle \exists Y.\text{woman}(Y) \Rightarrow (\forall X.\text{man}(X) \Rightarrow \text{loves}\ Y\ X), \emptyset\rangle
\end{array}$$

12.2 Type Raising for non-quantificational NPs

Proper names

- **Problem:** Subject NPs with quantificational determiners have type $(\iota \to o) \to o$ (and are applied to the VP) whereas subject NPs with proper names have type ι. (argument to the VP)
- **Idea:** *John runs* translates to $\mathrm{runs}(\mathrm{john})$, where $\mathrm{runs} \in \Sigma_{\iota \to o}$ and $\mathrm{john} \in \Sigma_{\iota}$.
  Now we $=_\beta$-expand over the VP yielding $(\lambda P_{\iota \to o}.P(\mathrm{john}))\ \mathrm{runs}$
  $\lambda P_{\iota \to o}.P(\mathrm{john})$ has type $(\iota \to o) \to o$ and can be applied to the VP runs.
- **Definition 2.1.** If $c \in \Sigma_\alpha$, then type raising c yields $\lambda P_{\alpha \to o}.P\ c$.

Definite NPs

- **Problem:** On our current assumptions, $\mathit{the}' = \iota$, and so for any definite NP *the N*, its translation is $\iota\ N$, an expression of type ι.
- **Idea:** Type lift just as we did with proper names: $\iota\ N$ type lifts to $\lambda P.P\ \iota\ N$, so $\mathit{the}' = \lambda PQ.Q\ \iota\ P$
- **Advantage:** This is a "generalized quantifier treatment": the' treated as denoting relations between sets.
- **Solution by Barwise&Cooper 1981:** For any $\mathrm{a} \in \mathcal{D}_{\iota\to o}$:
  $\mathcal{I}(\mathit{the}')(\mathrm{a}) = \mathcal{I}(\mathit{every}')(\mathrm{a})$ if $\#(\mathrm{a}) = 1$, undefined otherwise
  So the' is that function in $\mathcal{D}_{(\iota\to o)\to(\iota\to o)\to o}$ such that for any $A, B \in \mathcal{D}_{\iota\to o}$
  if $\#(A) = 1$ then $\mathit{the}'(A, B) = \mathsf{T}$ if $A \subseteq B$ and $\mathit{the}'(A, B) = \mathsf{F}$ if $A \not\subseteq B$ otherwise undefined

Problems with Type raised NPs

- **Problem:** We have type-raised NPs, but consider transitive verbs as in *Mary loves most cats*. loves is of type $\iota \to \iota \to o$ while the object NP is of type $(\iota \to o) \to o$ (application?)
- **Another Problem:** We encounter the same problem in the sentence *Mary loves John* if we choose to type-lift the NPs.
- **Idea:** Change the type of the transitive verb to allow it to “swallow” the higher-typed object NP.
- **Better Idea:** Adopt a new rule for semantic composition for this case.
- **Remember:** *loves′* is a function from individuals (e.g. *John*) to properties (in the case of the VP *loves John*, the property *X loves John* of X).

Type raised NPs and Function Composition

- We can extend $\mathrm{HOL}^{\rightarrow}$ by a constant $\circ_{(\beta\rightarrow\gamma)\rightarrow(\alpha\rightarrow\beta)\rightarrow\alpha\rightarrow\gamma}$ by setting $\circ := \lambda FGX.F(G(X))$ thus

$$\circ\; g\; f \rightarrow_{\beta} \lambda X.g(f(X)) \qquad \text{and} \qquad \circ\; g\; f\; a \rightarrow_{\beta} g(f(a))$$

In our example, we have

$$\begin{aligned}\circ\;(\lambda P.P(\mathrm{john}))\;\mathrm{loves} \quad &=_{Def} \quad (\lambda FGX.F(G(X)))\;(\lambda P.P(\mathrm{john}))\;\mathrm{loves}\\ &\rightarrow_{\beta} \quad (\lambda GX.(\lambda P.P(\mathrm{john}))\;G(X))\;\mathrm{loves}\\ &\rightarrow_{\beta} \quad \lambda X.(\lambda P.P(\mathrm{john}))\;\mathrm{loves}\;X\\ &\rightarrow_{\beta}! \quad \lambda X.\mathrm{loves}(X,\mathrm{john})\end{aligned}$$

12.3 Dealing with Quantifier Scope Ambiguity: Cooper Storage

Type raising transitive verbs

- We need transitive verbs to combine with quantificational objects of type $(\iota \to o) \to o$ but …
- We still ultimately want their "basic" translation to be type $\iota \to \iota \to o$, i.e. something that denotes a relation between individuals.
- We do this by starting with the basic translation, and raising its type. Here is what we'll end up with, for the verb *like*:

$$\lambda PY.P\,(\lambda X.\mathrm{likes}(X, Y))$$

where P is a variable of type $(\iota \to o) \to o$ and X, Y are variables of type ι. (For details on how this is derived, see [CKG09, pp.178-179])

Cooper Storage

- **Intuition:** A store consists of a “core” semantic representation, computed in the usual way, plus the representations of quantifiers encountered in the composition so far.
- **Definition 3.1.** A store is an n place sequence. The first member of the sequence is the core semantic representation. The other members of the sequence (if any) are pairs (β,i) where:
  - β is a QNP translation and
  - i is an index, which will associate the NP translation with a free variable in the core semantic translation.

  We call these pairs binding operators (because we will use them to bind free variables in the core representation).
- **Definition 3.2.** In the Cooper storage method, QNPs are stored in the store and later retrieved – not necessarily in the order they were stored – to build the representation.
- The elements in the store are written enclosed in angled brackets. However, we will often have a store which consists of only one element, the core semantic representation. This is because QNPs are the only things which add elements beyond the core representation to the store. So we will adopt the convention that when the store has only one element, the brackets are omitted.

How we put QNPs in the Store

- **Storage Rule**
  If the store $\langle\varphi,(\beta,j),\ldots,(\gamma,k)\rangle$ is a possible translation for a QNP, then the store
  $$\langle\lambda P.P(X_i)(\varphi,i)(\beta,j),\ldots,(\gamma,k)\rangle$$
  where i is a new index, is also a possible translation for that QNP.
- This rule says: if you encounter a QNP with translation φ, you can replace its translation with an indexed place holder of the same type, $\lambda P.P(X_i)$, and add φ to the store, paired with the index i. We will use the place holder translation in the semantic composition of the sentence.

Working with Stores

- Working out the translation for *Every student likes some professor.*

$$\begin{array}{lll}
NP_1 & \rightarrow & \lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)) \textbf{ or } \langle \lambda Q.Q(X_1), (\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)), 1)\rangle \\
V_t & \rightarrow & \lambda RY.R\ (\lambda Z.\mathrm{likes}(Z, Y)) \\
VP & \rightarrow & \text{(Combine core representations by FA; pass store up)*} \\
 & \rightarrow & \langle \lambda Y.\mathrm{likes}(X_1, Y), (\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)), 1)\rangle \\
NP_2 & \rightarrow & \lambda P.(\forall Z.\mathrm{student}(Z) \Rightarrow P(Z)) \textbf{ or } \langle \lambda R.R(X_2), (\lambda P.(\forall Z.\mathrm{student}(Z) \Rightarrow P(Z)), 2 \\
S & \rightarrow & \text{(Combine core representations by FA; pass stores up)**} \\
 & \rightarrow & \langle \mathrm{likes}(X_1, X_2), (\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)), 1), (\lambda P.(\forall Z.\mathrm{student}(Z) \Rightarrow P(Z)), 2
\end{array}$$

* Combining V_t with place holder

1. $(\lambda RY.R\ (\lambda Z.\mathrm{likes}(Z, Y)))\ (\lambda Q.Q(X_1))$
2. $\lambda Y.(\lambda Q.Q(X_1))\ (\lambda Z.\mathrm{likes}(Z, Y))$
3. $\lambda Y.(\lambda Z.\mathrm{likes}(Z, Y))\ X_1$
4. $\lambda Y.\mathrm{likes}(X_1, Y)$

** Combining VP with place holder

1. $(\lambda R.R(X_2))\ (\lambda Y.\mathrm{likes}(X_1, Y))$
2. $(\lambda Y.\mathrm{likes}(X_1, Y))\ X_2$
3. $\mathrm{likes}(X_1, X_2)$

Retrieving NPs from the store

- **Retrieval:**
  Let σ_1 and σ_2 be (possibly empty) sequences of binding operators. If the store $\langle\varphi, \sigma_1, \sigma_2, (\beta, i)\rangle$ is a translation of an expression of category S, then the store $\langle\beta(\lambda X_1.\varphi), \sigma_1, \sigma_2\rangle$ is also a translation of it.
- **What does this say?:** It says: suppose you have an S translation consisting of a core representation (which will be of type o) and one or more indexed QNP translations. Then you can do the following:
  1. Choose one of the QNP translations to retrieve.
  2. Rewrite the core translation, λ-abstracting over the variable which bears the index of the QNP you have selected. (Now you will have an expression of type $\iota \to o$.)
  3. Apply this λ-term to the QNP translation (which is of type $(\iota \to o) \to o$).

Example: *Every student likes some professor.*

1. Retrieve *every student*
   1.1 $(\lambda Q.(\forall Z.\mathrm{student}(Z) \Rightarrow Q(Z)))\ (\lambda X_2.\mathrm{likes}(X_1, X_2))$
   1.2 $\forall Z.\mathrm{student}(Z) \Rightarrow (\lambda X_2.\mathrm{likes}(X_1, X_2))\ Z$
   1.3 $\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X_1, Z)$
2. Retrieve *some professor*
   2.1 $(\lambda P.(\exists X.\mathrm{prof}(X) \wedge P(X)))\ (\lambda X_1.(\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X_1, Z)))$
   2.2 $\exists X.\mathrm{prof}(X)(\lambda X_1.(\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X_1, Z)))\ X$
   2.3 $\exists X.\mathrm{prof}(X) \wedge (\forall Z.\mathrm{student}(Z) \Rightarrow \mathrm{likes}(X, Z))$

Chapter 13

Higher-Order Unification and NL Semantics Reconstruction

13.1 Introduction

Application of HOL in NL Semantics: Ellipsis

- **Example 1.1.** *John loves his wife. George does too*
  - $\mathrm{loves}(\mathrm{john},\mathrm{wifeof}(\mathrm{john}))\wedge Q(\mathrm{george})$
  - "*George* has property some Q, which we still have to determine"
- **Idea:** If *John* has property Q, then it is that he *loves his wife*.
- **Equation:** $Q(\mathrm{john}) =_{\alpha\beta\eta} \mathrm{loves}(\mathrm{john},\mathrm{wifeof}(\mathrm{john}))$
- **Solutions (computed by HOU):**
  - $Q=\lambda z.\mathrm{loves}(z,\mathrm{wifeof}(z))$ and $Q=\lambda z.\mathrm{loves}(z,\mathrm{wifeof}(\mathrm{john}))$

  * $Q=\lambda z.\mathrm{loves}(\mathrm{john},\mathrm{wifeof}(z))$ and $Q=\lambda z.\mathrm{loves}(\mathrm{john},\mathrm{wifeof}(\mathrm{john}))$
- **Readings:** *George loves his own wife*. and *George loves John's wife*.
- **Erraneous HOU Predictions:** * *John loves George's wife*. and * *John loves John's wife*.

Higher-Order Unification (HOU)

- **Intuitively:** Equation solving in the simply typed λ-calculus (modulo the built-in $\alpha\beta\eta$-equality)
- **Formally:** Given formulae $\mathsf{A}, \mathsf{B} \in \mathit{wff}_{\alpha}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$, find a substitution σ with $\sigma(\mathsf{A}) =_{\alpha\beta\eta} \sigma(\mathsf{B})$.
- **Definition 1.2.**
  We call $\mathcal{E} := \mathsf{A}_1 =^? \mathsf{B}_1 \wedge \ldots \wedge \mathsf{A}_n =^? \mathsf{B}_n$ a unification problem. The set $\mathrm{U}(\mathcal{E}) = \{\sigma \mid \sigma(\mathsf{A}_i) =_{\alpha\beta\eta} \sigma(\mathsf{B}_i)\}$ is called the set of unifiers for $\mathcal{E}$ and any of its members a unifier.
- **Example 1.3.** the unification problem $F(fa) =^? f(Fa)$ where $F, f : \alpha \to \alpha$ and $\vdash_{\Sigma} a : \alpha$ has unifiers $[f/F], [\lambda X_{\alpha}.f(fX)/F], [\lambda X_{\alpha}.f(f(fX))/F], \ldots$
- find Representatives that induce all of $\mathrm{U}(\mathcal{E})$ (are there most general unifiers?)

Discourse Coherence

- Meaning of a discourse is more than just the conjunction of sentences
- Coherence is prerequisite for well-formedness (not just pragmatics)

  A *John killed Peter.*
  B^1 *No, John killed BILL!*
  B^2 * *No, John goes hiking!*
  B^3 *No, PETER died in that fight!*

- Coherence in a discourse is achieved by discourse relations
  - in this case "contrastive parallelism"

Discourse Relations (Examples)

- **Parallel:** *John organized rallies for Clinton, and Fred distributed pamphlets for him.*
- **Contrast:** *John supported Clinton, but Mary opposed him.*
- **Exemplification:** *Young aspiring politicians often support their party's presidential candidate. For instance John campaigned hard for Clinton in 1996.*
- **Generalization:** *John campaigned hard for Clinton in 1996. Young aspiring politicians often support their party's presidential candidate.*
- **Elaboration:** *A young aspiring politician was arrested in Texas today. John Smith, 34, was nabbed in a Houston law firm while attempting to embezzle funds for his campaign.*

Discourse Relations (The General Case)

- We need inferences to discover them
- General conditions [Hobbs 1990]

Relation	Requirements	Particle
Parallel	$a_i \sim b_i, p \mathrel{=\!\!\!|\!\!\!=} q$	*and*
Contrast	$a_i \sim b_i$, $p \models \neg q$ or $\neg p \models q$ a_i, b_i contrastive	*but*
Exempl.	$p \Dashv q$, $a_i \in \vec{b}$ or $a_i \Dashv b_i$	*for example*
Generl.	$p \Dashv q$, $b_i \in \vec{a}$ or $b_i \Dashv a_i$	*in general*
Elabor.	$q \simeq p$, $a_i \sim b_i$	*that is*

Source semantics $p(a_1, \ldots, a_n)$, Target semantics $q(a1, \ldots, a_m)$

- Need theorem proving methods for general case.

Underspecification/Ellipsis

- Natural language is economic
- Use the hearer's inferential capabilities to reduce communication costs.
- Makes use of discourse coherence for reconstruction (here: Parallelism)
  - *Jon loves his wife. Bill does too*. [love his/Bill's wife]
  - *Mary wants to go to Spain and Fred wants to go to Peru, but because of limited resources, only one of them can*. [go where he/she wants to go]
- Anaphora give even more coherence. (here: Elaboration)
  - *I have a new car. It is in the parking lot downstairs*. [My new car]
- Discourse relation determines the value of underspecified element.

Analyses based on Parallelism

- HOU Analyses (the structural level)
  - Ellipsis [DSP'91, G&K'96, DSP'96, Pinkal, et al'97]
  - Focus [Pulman'95, G&K96]
  - Corrections [G&K& v. Leusen'96]
  - Deaccenting, Sloppy Interpretation [Gardent, 1996]
- Discourse theories (the general case, needs deduction!)
  - Literature and Cognition [Hobbs, CSLI Notes'90]
  - Cohesive Forms [Kehler, PhD'95]
- **Problem:** All assume parallelism structure: given a pair of parallel utterances, the parallel elements are taken as given.

13.2 Higher-Order Unification

13.2.1 Higher-Order Unifiers

HOU: Complete Sets of Unifiers

- **Question:** Are there most general higher-order Unifiers?
- **Answer:** What does that mean anyway?
- **Definition 2.1.** $\sigma =_{\beta\eta} \rho[W]$, iff $\sigma(X) =_{\alpha\beta\eta} \rho(X)$ for all $X \in W$. $\sigma =_{\beta\eta} \rho[\mathcal{E}]$ iff $\sigma =_{\beta\eta} \rho[\mathrm{free}(\mathcal{E})]$
- **Definition 2.2.** σ is more general than θ on W ($\sigma \leq_{\beta\eta} \theta[W]$), iff there is a substitution ρ with $\theta =_{\beta\eta} (\rho \circ \sigma)[W]$.
- **Definition 2.3.** $\Psi \subseteq \mathrm{U}(\mathcal{E})$ is a complete set of unifiers, iff for all unifiers $\theta \in \mathrm{U}(\mathcal{E})$ there is a $\sigma \in \Psi$, such that $\sigma \leq_{\beta\eta} \theta[\mathcal{E}]$.
- **Definition 2.4.** If $\Psi \subseteq \mathrm{U}(\mathcal{E})$ is complete, then $\leq_{\beta}$-minimal elements $\sigma \in \Psi$ are most general unifier of $\mathcal{E}$.
- **Theorem 2.5.** *The set* $\{[\lambda uv.h\; u/F]\} \cup \{\sigma_i \mid i \in \mathbb{N}\}$ *where*

  $$\sigma_i := [\lambda uv.g_n\; u\; u\; h_1^n\; u\; v\; \ldots\; u\; h_n^n\; u\; v/F], [\lambda v.z/X]$$

  is a complete set of unifiers for the equation $F\; X\; (a_\iota) =^? F\; X\; (b_\iota)$, *where* F *and* X *are variables of types* $(\iota \rightarrow \iota) \rightarrow \iota \rightarrow \iota$ *and* $\iota \rightarrow \iota$
  Furthermore, σ_{i+1} *is more general than* σ_i.
- *Proof sketch:* [Hue76, Theorem 5]

Unification

- **Definition 2.6.** $X^1 =^? B^1 \wedge \ldots \wedge X^n =^? B^n$ is in solved form, if the X^i are distinct free variables $X^i \notin \mathrm{free}(B^j)$ and B^j does not contain Skolem constants for all j.
- **Lemma 2.7.** *If* $\mathcal{E} = X^1 =^? B^1 \wedge \ldots \wedge X^n =^? B^n$ *is in solved form, then* $\sigma_{\mathcal{E}} := [B^1/X^1], \ldots, [B^n/X^n]$ *is the unique most general unifier of* $\mathcal{E}$
- *Proof:*
  1. $\sigma(X^i) =_{\alpha\beta\eta} \sigma(B^i)$, so $\sigma \in \mathbf{U}(\mathcal{E})$
  2. Let $\theta \in \mathbf{U}(\mathcal{E})$, then $\theta(X^i) =_{\alpha\beta\eta} \theta(B^i) = \theta \circ \sigma(X^i)$
  3. so $\theta \leq_{\beta\eta} (\theta \circ \sigma)[\mathcal{E}]$.

13.2.2 Higher-Order Unification Transformations

Simplification $\mathcal{SIM}$

- **Definition 2.8.** The higher order simplification transformations $\mathcal{SIM}$ consist of the rules below.

$$\frac{(\lambda X_\alpha.A)=^?(\lambda Y_\alpha.B) \wedge \mathcal{E} \quad s \in \Sigma_\alpha^{Sk}\mathrm{new}}{([s/X](A))=^?([s/Y](B)) \wedge \mathcal{E}} \mathcal{SIM}{:}\alpha$$

$$\frac{(\lambda X_\alpha.A)=^?B \wedge \mathcal{E} \quad s \in \Sigma_\alpha^{Sk}\mathrm{new}}{([s/X](A))=^?Bs \wedge \mathcal{E}} \mathcal{SIM}{:}\eta$$

$$\frac{(h\,\overline{U^n})=^?(h\,\overline{V^n}) \wedge \mathcal{E} \quad h \in (\Sigma \cup \Sigma^{Sk})}{U_1=^?V_1 \wedge \ldots \wedge U_n=^?V_n \wedge \mathcal{E}} \mathcal{SIM}{:}\mathrm{dec}$$

$$\frac{\mathcal{E} \wedge X=^?A \quad X \notin \mathrm{free}(A) \quad A \cap \Sigma^{Sk} = \emptyset \quad X \in \mathrm{free}(\mathcal{E})}{[A/X](\mathcal{E}) \wedge X=^?A} \mathcal{SIM}{:}\mathrm{elim}$$

After rule applications all λ-terms are reduced to head normal form.

Properties of Simplification I

- **Lemma 2.9 (Properties of $\mathcal{SIM}$).** *$\mathcal{SIM}$ generalizes first-order unification.*
  - *$\mathcal{SIM}$ is terminating and confluent up to α-conversion*
  - *Unique $\mathcal{SIM}$ normal forms exist (all pairs have the form $(h\,\overline{\mathsf{U}^n}) =^? (k\,\overline{\mathsf{V}^m})$)*
- **Lemma 2.10.** $\mathsf{U}(\mathcal{E} \wedge \mathcal{E}_\sigma) = \mathsf{U}(\sigma(\mathcal{E}) \wedge \mathcal{E}_\sigma)$.
- *Proof:* by the definitions
  1. If $\theta \in \mathsf{U}(\mathcal{E} \wedge \mathcal{E}_\sigma)$, then $\theta \in (\mathsf{U}(\mathcal{E}) \cap \mathsf{U}(\mathcal{E}_\sigma))$.
  2. So $\theta =_{\beta\eta} (\theta \circ \sigma)[\mathrm{supp}(\sigma)]$,
  3. and thus $\theta \circ \sigma \in \mathsf{U}(\mathcal{E})$, iff $\theta \in \mathsf{U}(\sigma(\mathcal{E}))$.

Properties of Simplification II

- **Theorem 2.11.** *If* $\mathcal{E}\vdash_{\mathcal{SIM}}\mathcal{F}$*, then* $\mathrm{U}(\mathcal{E})\leq_{\beta\eta}\mathrm{U}(\mathcal{F})[\mathcal{E}]$*.* *(correct, complete)*

  Proof: By an induction over the length of the derivation

  We the $\mathcal{SIM}$ *rules individually for the base case*

  1. $\mathcal{SIM}{:}\alpha$ *by* $\alpha\beta\eta$*-conversion*
  2. $\mathcal{SIM}{:}\eta$ *By* η*-conversion in the presence of* $\mathcal{SIM}{:}\alpha$
  3. $\mathcal{SIM}{:}\mathrm{dec}$ *The head* $h\in(\Sigma\cup\Sigma^{Sk})$ *cannot be instantiated.*
  4. $\mathcal{SIM}{:}\mathrm{elim}$ *By* **??***.*
  5. The step case goes directly by induction hypothesis and transitivity of the derivation relation.

General Bindings

- **Problem:** Find all formulae of given type α and head h.
- **sufficient:** long $\beta\eta$ head normal form, most general.
- **Definition 2.12 (General Bindings).** $\mathrm{G}_{\alpha}^{h}(\Sigma) := \lambda\overline{X_{\alpha}^{k}}.h(H^{1}\ \overline{X})\ldots(H^{n}\ \overline{X})$
  - where $\alpha = \overline{\alpha}_{k} \to \beta$, $h:\overline{\gamma}_{n} \to \beta$ and $\beta \in \mathcal{BT}$
  - and $H^{i}:\overline{\alpha}_{k} \to \gamma_{i}$ new variables.

  is called the general binding of type α for the head h.
- **Observation 2.13.**
  General bindings are unique up to choice of names for H^{i}.
- **Definition 2.14.** If the head h is j^{th} bound variable in $\mathrm{G}_{\alpha}^{h}(\Sigma)$, call $\mathrm{G}_{\alpha}^{h}(\Sigma)$ j-projection binding (and write $\mathrm{G}_{\alpha}^{j}(\Sigma)$) else imitation binding
- clearly $\mathrm{G}_{\alpha}^{h}(\Sigma) \in \mathit{wff}_{\alpha}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ and $\mathrm{head}(\mathrm{G}_{\alpha}^{h}(\Sigma)) = h$

Approximation Theorem

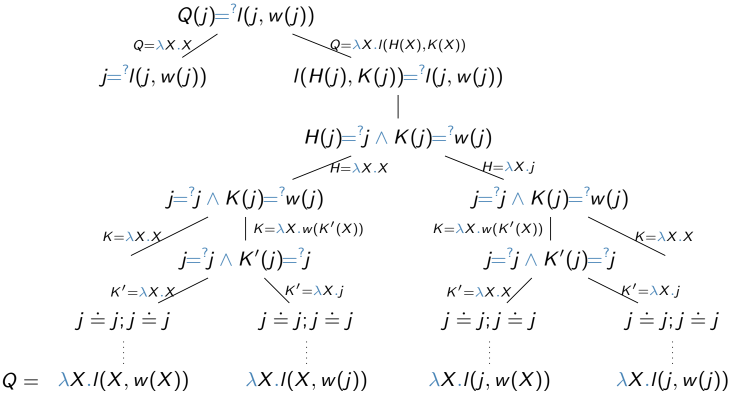
- **Theorem 2.15.** *If* A $\in wff_\alpha(\Sigma_\mathcal{T}, \mathcal{V}_\mathcal{T})$ *with* $\mathrm{head}(\mathrm{A}) = h$, *then there is a general binding* $\mathrm{G} = \mathrm{G}^h_\alpha(\Sigma)$ *and asubstitution* ρ *with* $\rho(\mathrm{G}) =_{\alpha\beta\eta} \mathrm{A}$ *and* $\mathrm{dp}\rho < \mathrm{dp}\mathrm{A}$*.*
- *Proof:* We analyze the term structure of A
  1. If $\alpha = \overline{\alpha}_k \to \beta$ and $h{:}\overline{\gamma}_n \to \beta$ where $\beta \in \mathcal{BT}$, then the long head normal form of A must be $\lambda \overline{X^k_\alpha}.h\,\overline{\mathrm{U}^n}$.
  2. $\mathrm{G} = \mathrm{G}^h_\alpha(\Sigma) = \lambda \overline{X^k_\alpha}.h(H_1\,\overline{X})\ldots(H_n\,\overline{X})$ for some variables $H_i{:}\overline{\alpha}_k \to \gamma_i$.
  3. Choose $\rho := [\lambda \overline{X^k_\alpha}.\mathrm{U}_1/H_1],\ldots,[\lambda \overline{X^k_\alpha}.\mathrm{U}_n/H_n]$.
  4. Then we have $$\begin{aligned}\rho(\mathrm{G}) &= \lambda \overline{X^k_\alpha}.h(\lambda \overline{X^k_\alpha}.\mathrm{U}_1\,\overline{X})\ldots(\lambda \overline{X^k_\alpha}.\mathrm{U}_n\,\overline{X})\\ &=_{\beta\eta} \lambda \overline{X^k_\alpha}.h\,\overline{\mathrm{U}^n}\\ &=_{\beta\eta} \mathrm{A}\end{aligned}$$
  5. The depth condition can be read off as $\mathrm{dp}(\lambda \overline{X^k_\alpha}.\mathrm{U}_1) \leq \mathrm{dp}\mathrm{A} - 1$.

Higher-Order Unification ($\mathcal{HOU}$)

- **Recap:** After simplification, we have to deal with pairs where one (flex/rigid) or both heads (flex/flex) are variables
- **Definition 2.16.** Let $\mathrm{G} = \mathrm{G}_\alpha^h(\Sigma)$ (imitation) or $\mathrm{G} \in \{\mathrm{G}_\alpha^j(\Sigma) \,|\, 1 \leq j \leq n\}$, then the calculus $\mathcal{HOU}$ for higher-order unification consists of the transformations (always reduce to $\mathcal{SIM}$ normal form)
  - Rule for flex/rigid pairs: $$\frac{(F_\alpha\ \overline{\mathrm{U}}) =^? (h\ \overline{\mathrm{V}}) \wedge \mathcal{E}}{F =^? \mathrm{G} \wedge (F\ \overline{\mathrm{U}}) =^? (h\ \overline{\mathrm{V}}) \wedge \mathcal{E}}\ \mathcal{HOU}:\mathrm{fr}$$
  - Rules for flex/flex pairs: $$\frac{(F_\alpha\ \overline{\mathrm{U}}) =^? (H\ \overline{\mathrm{V}}) \wedge \mathcal{E}}{F =^? \mathrm{G} \wedge (F\ \overline{\mathrm{U}}) =^? (H\ \overline{\mathrm{V}}) \wedge \mathcal{E}}\ \mathcal{HOU}:\mathrm{ff}$$

$\mathcal{HOU}$ Example

Example 2.17. Let $Q, w{:}\iota \to \iota$, $l{:}\iota \to \iota \to \iota$, and $j{:}\iota$, then we have the following derivation tree in $\mathcal{HOU}$.



$Q(j) =^? l(j, w(j))$

$Q = \lambda X.X$ | $Q = \lambda X.l(H(X), K(X))$

$j =^? l(j, w(j))$ | $l(H(j), K(j)) =^? l(j, w(j))$

$H(j) =^? j \wedge K(j) =^? w(j)$

$H = \lambda X.X$ | $H = \lambda X.j$

$j =^? j \wedge K(j) =^? w(j)$ | $j =^? j \wedge K(j) =^? w(j)$

$K = \lambda X.X$ | $K = \lambda X.w(K'(X))$ | $K = \lambda X.w(K'(X))$ | $K = \lambda X.X$

$j =^? j \wedge K'(j) =^? j$ | $j =^? j \wedge K'(j) =^? j$

$K' = \lambda X.X$ | $K' = \lambda X.j$ | $K' = \lambda X.X$ | $K' = \lambda X.j$

$j \doteq j; j \doteq j$ | $j \doteq j; j \doteq j$ | $j \doteq j; j \doteq j$ | $j \doteq j; j \doteq j$

$Q = \lambda X.l(X, w(X))$ | $\lambda X.l(X, w(j))$ | $\lambda X.l(j, w(X))$ | $\lambda X.l(j, w(j))$

A Test Generator for Higher-Order Unification

- **Definition 2.18 (Church Numerals).** We define closed λ-terms of type $\nu := (\alpha \to \alpha) \to \alpha \to \alpha$
  - Numbers: Church numerals: (*n* fold iteration of arg1 starting from arg2)

$$n := \lambda S_{\alpha \to \alpha}.\lambda O_{\alpha}.\underbrace{S(S \ldots S}_{n}(O) \ldots)$$

  - Addition (*N*-fold iteration of *S* from *N*)

$$+ := \lambda N_{\nu} M_{\nu}.\lambda S_{\alpha \to \alpha}.\lambda O_{\alpha}.NS(MSO)$$

  - Multiplication: (*N*-fold iteration of *MS* (=+*m*) from *O*)

$$\cdot := \lambda N_{\nu} M_{\nu}.\lambda S_{\alpha \to \alpha}.\lambda O_{\alpha}.N(MS)O$$

- **Observation 2.19.** *Subtraction and (integer) division on Church numberals can be automted via higher-order unification.*
- **Example 2.20.**
  $5 - 2$ by solving the unification problem $(2 + x_{\nu}) =^{?} 5$
- Equation solving for Church numerals yields a very nice generator for test cases for higher-order unification, as we know which solutions to expect.

13.2.3 Properties of Higher-Order Unification

Undecidability of Higher-Order Unification

- **Theorem 2.21.** *Second-order unification is undecidable* *(Goldfarb '82 [Gol81])*
- *Proof sketch:* Reduction to Hilbert's tenth problem (solving Diophantine equations) (known to be undecidable)
- **Definition 2.22.**
  We call an equation a Diophantine equation, if it is of the form
  - $x_i x_j = x_k$
  - $x_i + x_j = x_k$
  - $x_i = c_j$ where $c_j \in \mathbb{N}$

  where the variables x_i range over $\mathbb{N}$.
- These can be solved by higher-order unification on Church numerals. (cf. **??**).
- **Theorem 2.23.** *The general solution for sets of Diophantine equations is undecidable.* *(Matijasevič 1970 [Mat70])*

$\mathcal{HOU}$ is Correct

- **Lemma 2.24.** *If* $\mathcal{E}\vdash_{\mathcal{HOU}\mathsf{fr}}\mathcal{E}'$ *or* $\mathcal{E}\vdash_{\mathcal{HOU}\mathsf{ff}}\mathcal{E}'$*, then* $\mathrm{U}(\mathcal{E}')\subseteq\mathrm{U}(\mathcal{E})$.
- *Proof sketch:* $\mathcal{HOU}$:fr and $\mathcal{HOU}$:ff only add new pair.
- **Corollary 2.25.** $\mathcal{HOU}$ *is correct: If* $\mathcal{E}\vdash_{\mathcal{HOU}}\mathcal{E}'$*, then* $\mathrm{U}(\mathcal{E}')\subseteq\mathrm{U}(\mathcal{E})$.

Completeness of $\mathcal{HOU}$

- We cannot expect completeness in the same sense as for first-order unification: "If $\mathcal{E}\vdash_{\mathcal{U}}\mathcal{F}$, then $\mathrm{U}(\mathcal{E})\subseteq\mathrm{U}(\mathcal{F})$" (see **??**) as the rules fix a binding and thus partially commit to a unifier (which excludes others).
- We cannot expect termination either, since HOU is undecidable.
- For a semi-decision procedure we only need termination on unifiable problems.
- **Theorem 2.26 ($\mathcal{HOU}$ derives Complete Set of Unifiers).** *If $\theta\in\mathrm{U}(\mathcal{E})$, then there is a $\mathcal{HOU}$-derivation $\mathcal{E}\vdash_{\mathcal{HOU}}\mathcal{F}$, such that $\mathcal{F}$ is in solved form, $\sigma_{\mathcal{F}}\in\mathrm{U}(\mathcal{E})$, and $\sigma_{\mathcal{F}}$ is more general than θ.*
- *Proof sketch:* Given a unifier θ of $\mathcal{E}$, we guide the derivation with a measure μ_θ towards $\mathcal{F}$.

Completeness of $\mathcal{HOU}$ (Measure)

- **Definition 2.27.** We call $\mu(\mathcal{E},\theta):=\langle\mu_1(\mathcal{E},\theta),\mu_2(\theta)\rangle$ the unification measure for $\mathcal{E}$ and θ, if
  - $\mu_1(\mathcal{E},\theta)$ is the multiset of term depths of $\theta(X)$ for the unsolved $X\in\mathrm{supp}(\theta)$.
  - $\mu_2(\mathcal{E})$ the multiset of term depths in $\mathcal{E}$.
  - $\prec$ is the strict lexicographic order on pairs: ($\langle a,b\rangle\prec\langle c,d\rangle$, if $a<c$ or $a=c$ and $b<d$)
  - Component orderings are multiset orderings: ($M\cup\{m\}<M\cup N$ iff $n<m$ for all $n\in N$)
- **Lemma 2.28.** *$\prec$ is well-founded.* *(by construction)*

Completeness of $\mathcal{HOU}$ (μ-Prescription)

- **Theorem 2.29.** *If $\mathcal{E}$ is unsolved and $\theta \in \mathrm{U}(\mathcal{E})$, then there is a unification problem $\mathcal{E}$ with $\mathcal{E} \vdash_{\mathcal{HOU}} \mathcal{E}'$ and a substitution $\theta' \in \mathrm{U}(\mathcal{E}')$, such that*
  - $\theta =_{\beta\eta} \theta'[\mathcal{E}]$
  - $\mu(\mathcal{E}, \theta' 0) \prec \mu(\mathcal{E}, \theta')$.

  we call such a $\mathcal{HOU}$-step a μ-prescribed
- **Corollary 2.30.** *If $\mathcal{E}$ is unifiable without μ-prescribed $\mathcal{HOU}$-steps, then $\mathcal{E}$ is solved.*
- **In other words:** μ guides the $\mathcal{HOU}$-transformations to a solved form.

Proof of **??** I

- *Proof:*
  1. Let $\mathrm{A} =^{?} \mathrm{B}$ be an unsolved pair of the form $(\mathrm{F}\ \overline{\mathrm{U}}) =^{?} (\mathrm{G}\ \overline{\mathrm{V}})$ in $\mathcal{F}$.
  2. $\mathcal{E}$ is a $\mathcal{SIM}$ normal form, so F and G must be constants or variables,
  3. but not the same constant, since otherwise $\mathcal{SIM}$:dec would be applicable.
  4. By **??** there is a general binding $\mathrm{G} = \mathrm{G}_{\alpha}^{f}(\Sigma)$ and a substitution ρ with $\rho(\mathrm{G}) =_{\alpha\beta\eta} \theta(F)$. So,
     - if $\mathrm{head}(\mathrm{G}) \notin \mathrm{supp}(\theta)$, then $\mathcal{HOU}$:fr is applicable,
     - if $\mathrm{head}(\mathrm{G}) \in \mathrm{supp}(\theta)$, then $\mathcal{HOU}$:ff is applicable.
  5. Choose $\theta' := \theta \cup \rho$. Then $\theta =_{\beta\eta} \theta'[\mathcal{E}]$ and $\theta' \in \mathrm{U}(\mathcal{E}')$ by correctness.
  6. $\mathcal{HOU}$:ff and $\mathcal{HOU}$:fr solve $F \in \mathrm{supp}(\theta)$ and replace F by $\mathrm{supp}(\rho)$ in the set of unsolved variable of $\mathcal{E}$.
  7. so $\mu_1(\mathcal{E}, \theta') \prec \mu_1(\mathcal{E}, \theta)'$ and thus $\mu(\mathcal{E}, \theta') \prec \mu(\mathcal{E}, \theta')$.

Terminal $\mathcal{HOU}$-problems are Solved or Unsolvable I

- **Theorem 2.31.** *If $\mathcal{E}$ is a unsolved UP and $\theta \in \mathrm{U}(\mathcal{E})$, then there is a $\mathcal{HOU}$-derivation $\mathcal{E} \vdash_{\mathcal{HOU}} \sigma_\sigma$, with $\sigma \leq_{\beta\eta} \theta[\mathcal{E}]$.*
- *Proof:* Let $\mathcal{D}\colon \mathcal{E} \vdash_{\mathcal{HOU}} \mathcal{F}$ a maximal μ-prescribed $\mathcal{HOU}$-derivation from $\mathcal{E}$.
  1. This must be finite, since $\prec$ is well-founded (ind. over length n of $\mathcal{D}$)
  2. If $n = 0$, then $\mathcal{E}$ is solved and $\sigma_\mathcal{E}$ most general unifier
  3. thus $\sigma_\mathcal{E} \leq_{\beta\eta} \theta[\mathcal{E}]$
  4. If $n > 0$, then there is a μ-prescribed step $\mathcal{E} \vdash_{\mathcal{HOU}} \mathcal{E}'$ and a substitution θ as in **??**.
  5. by IH there is a $\mathcal{HOU}$-derivation $\mathcal{E}' \vdash_{\mathcal{HOU}} \mathcal{F}$ with $\sigma_\mathcal{F} \leq_{\beta\eta} \theta'[\mathcal{E}']$.
  6. by correctness $\sigma_\mathcal{F} \in \mathrm{U}(\mathcal{E}') \subseteq \mathrm{U}(\mathcal{E})$.
  7. rules of $\mathcal{HOU}$ only expand free variables, so $\sigma_\mathcal{F} \leq_{\beta\eta} \theta'[\mathcal{E}']$.
  8. Thus $\sigma_\mathcal{F} \leq_{\beta\eta} \theta'[\mathcal{E}]$,
  9. This completes the proof, since $\theta' =_{\beta\eta} \theta[\mathcal{E}]$ by **??**.

Properties of HO-Unification

- HOU is undecidable,
- HOU need not have most general unifiers
- The $\mathcal{HOU}$ transformation induce an algorithm that enumerates a complete set of higher-order unifiers.
- $\mathcal{HOU}$:ff gives enormous degree of indeterminism
- HOU is intractable in practice consider restricted fragments where it is!
- HO Matching (decidable up to order four), HO Patterns (unitary, linear), . . .

13.2.4 Pre-Unification

Pre-Unification

- $\mathcal{HOU}$:ff has a giant branching factor in the search space for unifiers. (makes HOU impracticable)
- In most situations, we are more interested in solvability of unification problems than in the unifiers themselves.
- More liberal treatment of flex/flex pairs.
- **Observation 2.32.** *flex/flex-pairs* $(F\ \overline{\mathrm{U}^n})=^?(G\ \overline{\mathrm{V}^m})$ *are always (trivially) solvable by* $[\lambda\overline{X^n}.H/F],[\lambda\overline{Y^m}.H/G]$, *where* H *is a new variable*
- **Idea:** consider flex/flex-pairs as pre solved.
- **Definition 2.33 (Pre-Unification).** For given terms $\mathrm{A},\mathrm{B}\in wf\!f_\alpha(\Sigma_\mathcal{T},\mathcal{V}_\mathcal{T})$ find a substitution σ, such that $\sigma(\mathrm{A})=^p_{\beta\eta}\sigma(\mathrm{B})$, where $=^p_{\beta\eta}$ is the equality theory that is induced by $=_{\beta\eta}$ and $F\ \overline{\mathrm{U}}=G\ \overline{\mathrm{V}}$.
- **Lemma 2.34.** *A higher-order unification problem is unifiable, iff it is pre-unifiable.*

Pre-Unification Algorithm $\mathcal{HOPU}$

- **Definition 2.35.** A unification problem is a pre solved form, iff all of its pairs are solved or flex/flex
- **Lemma 2.36.** *If $\mathcal{E}$ is solved and $\mathcal{P}$ flex/flex, then σ_σ is a most general unifier of a pre-solved form $\mathcal{E} \wedge \mathcal{P}$.*
- Restrict all $\mathcal{HOU}$ rule so that they cannot be applied to pre-solved pairs.
- In particular, remove $\mathcal{HOU}$:ff!
- **Definition 2.37.** The higher-order pre-unification calculus $\mathcal{HOPU}$ only consists of $\mathcal{SIM}$ and $\mathcal{HOU}$:fr.
- **Theorem 2.38.** *$\mathcal{HOPU}$ is a correct and complete pre-unification algorithm*
- *Proof sketch:* with exactly the same methods as higher-order unification
- **Theorem 2.39.** *Higher-order pre-unification is infinitary, i.e. a unification problem can have infinitely many unifiers.* (Huet 76' [Hue76])
- **Example 2.40.** $Y\ (\lambda X_\iota.X)\ a =^? a$, where a is a constant of type ι and Y a variable of type $(\iota \to \iota) \to \iota \to \iota$ has the most general unifiers $\lambda sz.s^n\ z$ and $\lambda sz.s^n\ a$, which are mutually incomparable and thus most general.

13.2.5 Applications of Higher-Order Unification

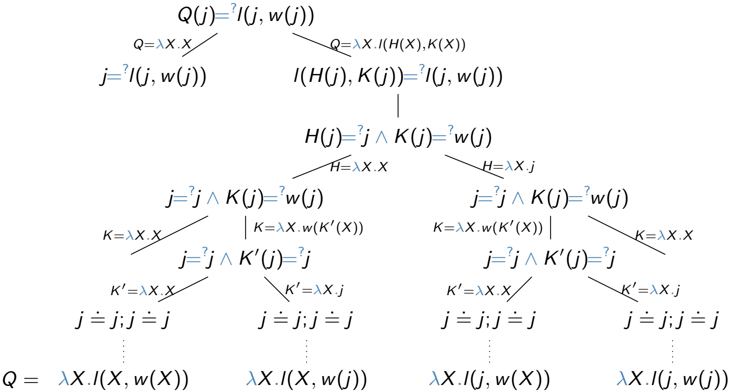
Application of HOL in NL Semantics: Ellipsis

- **Example 2.41.** *John loves his wife. George does too*
  - $\mathrm{loves}(\mathrm{john}, \mathrm{wifeof}(\mathrm{john})) \wedge Q(\mathrm{george})$
  - "*George* has property some Q, which we still have to determine"
- **Idea:** If *John* has property Q, then it is that he *loves his wife*.
- **Equation:** $Q(\mathrm{john}) =_{\alpha\beta\eta} \mathrm{loves}(\mathrm{john}, \mathrm{wifeof}(\mathrm{john}))$
- **Solutions (computed by HOU):**
  - $Q = \lambda z.\mathrm{loves}(z, \mathrm{wifeof}(z))$ and $Q = \lambda z.\mathrm{loves}(z, \mathrm{wifeof}(\mathrm{john}))$

  * $Q = \lambda z.\mathrm{loves}(\mathrm{john}, \mathrm{wifeof}(z))$ and $Q = \lambda z.\mathrm{loves}(\mathrm{john}, \mathrm{wifeof}(\mathrm{john}))$
- **Readings:** *George loves his own wife*. and *George loves John's wife*.
- **Erraneous HOU Predictions:** * *John loves George's wife*. and * *John loves John's wife*.

13.3 Linguistic Applications of Higher-Order Unification

George does too (HOU)



$Q(j) =^{?} l(j, w(j))$

$Q = \lambda X.X$

$Q = \lambda X.l(H(X), K(X))$

$j =^{?} l(j, w(j))$

$l(H(j), K(j)) =^{?} l(j, w(j))$

$H(j) =^{?} j \wedge K(j) =^{?} w(j)$

$H = \lambda X.X$

$H = \lambda X.j$

$j =^{?} j \wedge K(j) =^{?} w(j)$

$j =^{?} j \wedge K(j) =^{?} w(j)$

$K = \lambda X.X$

$K = \lambda X.w(K'(X))$

$K = \lambda X.w(K'(X))$

$K = \lambda X.X$

$j =^{?} j \wedge K'(j) =^{?} j$

$j =^{?} j \wedge K'(j) =^{?} j$

$K' = \lambda X.X$

$K' = \lambda X.j$

$K' = \lambda X.X$

$K' = \lambda X.j$

$j \doteq j; j \doteq j$

$j \doteq j; j \doteq j$

$j \doteq j; j \doteq j$

$j \doteq j; j \doteq j$

$Q = \quad \lambda X.l(X, w(X)) \qquad \lambda X.l(X, w(j)) \qquad \lambda X.l(j, w(X)) \qquad \lambda X.l(j, w(j))$

Primary Occurrence Restriction

- **Problem:** HOU over-generates
- **Idea:** [Dalrymple, Shieber, Pereira]

  Given a labeling of occurrences as either primary or secondary, the POR excludes of the set of linguistically valid solutions, any solution which contains a primary occurrence.
- A primary occurrence is an occurrence that is directly associated with a source parallel element.
- a source parallel element is an element of the source (i.e. antecedent) clause which has a parallel counterpart in the target (i.e. elliptic) clause.
- **Example 3.1.**
  - $\text{loves}(\underline{\text{john}}, \text{wifeof}(\text{john})) = Q(\text{george})$
  - $Q = \lambda x.\text{loves}(x, \text{wifeof}(\text{john}))$
  - $Q = \lambda x.\text{loves}(\underline{\text{john}}, \text{wifeof}(\text{john}))$
- Use the colored λ-calculus for general theory

Colored λ-calculus [HK00]

- Developed for inductive theorem proving (Rippling with Metavariable)
- **Definition 3.2.** Symbol occurrences can be annotated with colors (variables α, β, γ, …and constants a, b,…)
- Bound variables are uncolored ($\beta\eta$ conversion just as usual)
- **Definition 3.3.** Well-colored substitutions σ
  - Map colored variables X_X to colored formulae.
  - If a and b are different colors, then $|\sigma(X_X)| = |\sigma(X_X)|$: equal color erasures. (Consistency)
  - All color annotations on $\sigma(X_X)$ have to be compatible with those for c. (Monochromacity)

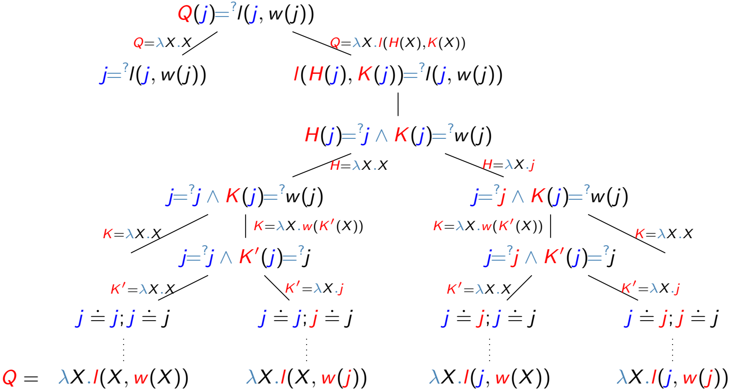
Colored HO-Unification

- HOCU has only two differences wrt. general HOU

$$\frac{f_f(t^1,\ldots,t^n) =^? f_f(s^1,\ldots,s^n)}{a =^? b \wedge t^1 =^? s^1 \wedge t^n =^? s^n} \qquad \frac{X_X =^? \mathrm{A} \wedge \mathcal{E}}{X =^? \mathrm{A} \wedge [\mathrm{A}/X](\mathcal{E})}$$

- Decomposition must consider colors
- Elimination ensures Monochromicity and Consistency
  - $X =^? \mathrm{A} := X_X =^? \mathrm{A}_\mathrm{A} \wedge X_X =^? \mathrm{A}_\mathrm{A}$
  - $[\mathrm{A}/X] := [\mathrm{A}_\mathrm{A}/X_X],\ldots,[\mathrm{A}_\mathrm{A}/X_X]$ propagates color information

George does too (HOCU)



$Q(j) \stackrel{?}{=} I(j, w(j))$

$Q = \lambda X.X$

$Q = \lambda X.I(H(X), K(X))$

$j \stackrel{?}{=} I(j, w(j))$

$I(H(j), K(j)) \stackrel{?}{=} I(j, w(j))$

$H(j) \stackrel{?}{=} j \wedge K(j) \stackrel{?}{=} w(j)$

$H = \lambda X.X$

$H = \lambda X.j$

$j \stackrel{?}{=} j \wedge K(j) \stackrel{?}{=} w(j)$

$j \stackrel{?}{=} j \wedge K(j) \stackrel{?}{=} w(j)$

$K = \lambda X.X$

$K = \lambda X.w(K'(X))$

$K = \lambda X.w(K'(X))$

$K = \lambda X.X$

$j \stackrel{?}{=} j \wedge K'(j) \stackrel{?}{=} j$

$j \stackrel{?}{=} j \wedge K'(j) \stackrel{?}{=} j$

$K' = \lambda X.X$

$K' = \lambda X.j$

$K' = \lambda X.X$

$K' = \lambda X.j$

$j \doteq j; j \doteq j$ $\quad$ $j \doteq j; j \doteq j$ $\quad$ $j \doteq j; j \doteq j$ $\quad$ $j \doteq j; j \doteq j$

$Q = \quad \lambda X.I(X, w(X)) \quad \lambda X.I(X, w(j)) \quad \lambda X.I(j, w(X)) \quad \lambda X.I(j, w(j))$

The Original Motivation: First-Order Rippling

Example 3.4. Proving: $\forall x, y : \mathit{list}.\mathrm{rev}(\mathrm{app}(\mathrm{rev}(x), y)) = \mathrm{app}(\mathrm{rev}(y), x)$

$$\begin{tikzcd}
\mathrm{rev}(\mathrm{app}(\mathrm{rev}(\mathrm{cons}(h, x)), y)) = \mathrm{app}(\mathrm{rev}(y), \mathrm{cons}(h, x)) \arrow[d] \\
\mathrm{rev}(\mathrm{app}(\mathrm{app}(\mathrm{rev}(x), \mathrm{cons}(h, \mathrm{nil})), y)) = \mathrm{app}(\mathrm{rev}(y), \mathrm{cons}(h, x)) \arrow[d, swap, "\mathrm{app}_{\alpha}(X_X, \mathrm{cons}(Y, Z_Z)) = \mathrm{app}\alpha(F_1(X_X, Y, Z), Z_Z)"] \\
\mathrm{rev}(\mathrm{app}(\mathrm{app}(\mathrm{rev}(x), \mathrm{cons}(h, \mathrm{nil})), y)) = \mathrm{app}(F_1(\mathrm{rev}(y), h, x), x) \arrow[d, swap, "\mathrm{app}(\mathrm{rev}_{\alpha}(Y_Y), \mathrm{cons}(X, \mathrm{nil})) = \mathrm{rev}_{\alpha}(\mathrm{cons}(X, Y_Y))"] \\
\mathrm{rev}(\mathrm{app}(\mathrm{app}(\mathrm{rev}(x), \mathrm{cons}(h, \mathrm{nil})), y)) = \mathrm{app}(\mathrm{rev}(\mathrm{cons}(h, y)), x) \arrow[d] \\
\mathrm{rev}(\mathrm{app}(\mathrm{rev}(x), \mathrm{cons}(h, y))) = \mathrm{app}(\mathrm{rev}(\mathrm{cons}(h, y)), x)
\end{tikzcd}$$

The Higher-Order Case: Schematic Rippling

Example 3.5 (Synthesizing Induction Orderings). $\forall x.\exists y.f(g(y)) \leq x$

Induction Step: $\forall x.\exists y.f(g(y)) \leq x$ to $\exists y.f(g(y)) \leq F(x)$

$$\begin{array}{rcll} f(g(y)) & \leq & F(x) & \\ f(s(g(y'))) & \leq & F(x) & \\ s(s(f(g(y')))) & \leq & F(x) & \\ s(s(f(g(y')))) & \leq & s(s(x)) & F \leftarrow \lambda X.s(s(X)) \\ f(g(y')) & \leq & x & \end{array}$$

A Unification Problem

- **Example 3.6 (A Unification Problem).**

$$\begin{tikzcd}
F(\mathrm{rev}(y), h, x) =^? rev_\alpha(Y_\beta)\ \mathrm{cons}(X, \mathrm{nil}) \arrow[d, "{[\lambda UVW.\mathrm{app}(H(U,V,W), K(U,V,W))/F]}"] \\
H(\mathrm{rev}(u), h, v) =^? \mathrm{rev}_\alpha(Y_Y) \wedge K(\mathrm{rev}(u), h, v) =^? \mathrm{cons}(X, \mathrm{nil}) \arrow[d, "{\begin{array}{c}[\lambda UVW.\mathrm{cons}(M(U,V,W), N(U,V,W))/K], \\ {[\lambda UVW.U/H]}\end{array}}"] \\
\mathrm{rev}(u) =^? \mathrm{rev}_\alpha(Y_Y) \wedge \mathrm{cons}(M(\mathrm{rev}(u), h, v), N(\mathrm{rev}(u), h, v)) =^? \mathrm{cons}(X, \mathrm{nil}) \arrow[d] \\
\alpha =^? \blacksquare \wedge u =^? Y_Y \wedge X =^? M(\mathrm{rev}(u), h, v) \wedge N(\mathrm{rev}(u), h, v) =^? nil \arrow[d] \\
h =^? h \wedge \mathrm{nil} =^? \mathrm{nil}
\end{tikzcd}$$

Result: $[\lambda UVW.\mathrm{app}(U, \mathrm{cons}(V, \mathrm{nil}))/F], [u/Y_Y], [h/X], [\blacksquare/\alpha]$

Linguistic Application: Focus/Ground Structures

- **Example 3.7.** *John only likes MARY.*
- **Analysis:** $\mathrm{likes}(\mathrm{john}, \mathrm{mary}) \wedge (\forall x.G(x)) \Rightarrow x = \mathrm{mary}$.
- **Equation:** $\mathrm{likes}(\mathrm{john}, \mathrm{mary}) =_{\alpha\beta\eta} G(\mathrm{mary})$.
  - Variable G for (back)ground (Focus is prosodically marked)
- **Solution:** $G = \lambda z.\mathrm{likes}(\mathrm{john}, z)$
- **Semantics:** $\mathrm{likes}(\mathrm{john}, \mathrm{mary}) \wedge (\forall x.\mathrm{likes}(\mathrm{john}, x) \Rightarrow x = \mathrm{mary})$.
- **Linguistic Coverage:** Prosodically unmarked focus, sentences with multiple focus operators
  [Gardent & Kohlhase'96]

Isn't HOCU just a notational variant of DSP's POR?

- HOCU has a *formal*, well–understood foundation which permits a clear assessment of its mathematical and computational properties;
- It is a *general* theory of colors:
- Other Constraints
  - POR for focus
  - Second Occurrence Expressions
  - Weak Crossover Constraints
- Multiple constraints and their interaction are easily handled
  - Use feature constraints as colors

Interaction of Constraints via Extended Colors

- **Example 3.8.** *John likes MARY and Peter does too*
  - Ellipsis: $l(j_j, s_s) = R_R(j_j)$
  - Focus: $R_R(p) = G_G(F_F)$
  - $\neg$pe forbids only pe $\neg$pf forbids only pf
- **Derivation:**
  - Solution $R_R = \lambda x.l(x, s_s)$ to the Ellipsis equation
  - yields Focus equation $l(p, s_s) = G_G(F_F)$
- **Solution:** $G_G = \lambda x.l(p_p, x)$ $F_F = m_m$

Featuring even more colors for Interaction

- *John$_1$'s mum loves him$_1$. Peter's mum does too.*
- Two readings:
  - *Peter's mum loves Peter* (sloppy)
  - *Peter's mum loves John* (strict)
- Parallelism equations

$$\begin{aligned} C(j) &= l(m(j), j) \\ C(p) &= R(m(p)) \end{aligned}$$

- Two solution for the first equation:

  $C = \lambda Z.l(m(Z), j)$ (strict) and $C = \lambda Z.l(m(Z), Z)$ (sloppy)
- Two versions of the second equation

$$\begin{aligned} l(m(p), j) &= R(m(p)) \\ l(m(p), p) &= R(m(p)) \end{aligned}$$

- $R = \lambda Z.l(Z, j)$ solves the first equation (strict reading)
- the second equation is unsolvable $R = \lambda Z.l(Z, p)$ is not well-colored.
- **Idea:** Need additional constraint:

  VPE may not contain (any part of) it's subject
- Need more dimensions of colors to model the interaction
- **Idea:** Extend supply of colors to feature terms.

John$_1$'s mum loves him$_1$. Peter's mum does too.

- Parallelism Constraints

$$\begin{aligned} C_C(j_j) &= l(m_m(j_j), j) \\ C_C(p_p) &= R_R(m_m(p_p)) \end{aligned}$$

- Resolving the first equation yields two possible values for C_C:

$$\lambda z.l(m_m(z), j) \quad \text{and} \quad \lambda z.l(m_m(z), z)$$

- Two versions of the second equation

$$\begin{aligned} l(m_m(p_p), j) &= R_R(m_m(p_p)) \\ l(m_m(p_p), p_p) &= R_R(m_m(p_p)) \end{aligned}$$

- Two solutions for the ellipsis (for R_R)

$$\begin{array}{ll} \{R_R \leftarrow \lambda z.l(z, j)\} & \text{Strict Reading} \\ \{R_R \leftarrow \lambda z.l(z, p_p)\} & \text{Sloppy Reading} \end{array}$$

- Need *dynamic constraints/*
  - resulting from the unification of several independent constraints
  - VPE subject is $[e \quad +]$, while part of is a parallel element ($[p \quad +]$).
  - Various linguistic modules interact in creating complex constraints

Computation of Parallelism (The General Case)

- We need inferences to discover discourse relations
- General Conditions [Hobbs 1990]

Relation	Requirements	Particle
Parallel	$a_i \sim b_i,\ p \simeq q$	*and*
Contrast	$a_i \sim b_i,\ p \supset \neg q$ or $\neg p \supset q$ a_i, b_i contrastive	*but*

Source semantics $p(\vec{a})$, Target semantics $q(\vec{b})$

- $a \sim b$, iff $\forall p.p(a) \Rightarrow (\exists q \simeq p.q(b))$ $p \simeq q$, iff $\forall a.p(a) \Rightarrow (\exists b \sim a.q(b))$
- Need theorem proving methods for general case.
- **Idea:** use only special properties (Sorts from the Taxonomy)

13.4 Sorted Higher-Order Unification

Sorted λ-Calculus

- higher-order automated theorem provers are relatively weak
- transfer first-order theorem proving technology to higher-order
- sorts are a particularly efficient refinement
  - separation of sorts and types
  - functional base sorts
  - term declarations as very general mechanism for declaring sort information

Sorted Unification:

- Example: Signature Σ with

$$\begin{array}{l}[+:(\mathbb{N}\to\mathbb{N}\to\mathbb{N})]\\ [+:(\mathbb{E}\to\mathbb{E}\to\mathbb{E})]\\ [+:(\mathbb{O}\to\mathbb{O}\to\mathbb{E})]\\ [(\lambda X.+XX):(\mathbb{N}\to\mathbb{E})]\end{array}$$

- general bindings

$$\mathrm{G}_{\mathbb{E}}^{+}()=\left\{\begin{array}{l}+Z_{\mathbb{E}}W_{\mathbb{E}},\\ +Z_{\mathbb{O}}W_{\mathbb{O}},\\ +Z_{\mathbb{N}}Z_{\mathbb{N}}\end{array}\right\}$$

Example (Elementary Calculus)

- Sorts
  - $\mathbb{R}^+$, $\mathbb{R}$ of type ι: (non-negative) real numbers
  - $\mathbb{M}$, $\mathbb{P}$ of type $\iota\to\iota$: monomials, polynomials
  - $\mathbb{M}$, $\mathbb{P}$ of type $\iota\to\iota$: differentiable and continuous functions
- Signature Σ

$$\begin{array}{l}
[+:(\mathbb{R}\to\mathbb{R}\to\mathbb{R})], [*:(\mathbb{R}\to\mathbb{R}\to\mathbb{R})], [(\lambda X.*XX):(\mathbb{R}\to\mathbb{R}^+)],\\
[\mathbb{R}^+\sqsubset\mathbb{R}], [\mathbb{M}\sqsubset\mathbb{P}], [\mathbb{P}\sqsubset\mathbb{M}], [\mathbb{M}\sqsubset\mathbb{P}]\\
[(\lambda X.X):\mathbb{M}], [(\lambda XY.Y):(\mathbb{R}\to\mathbb{M})],\\
[(\lambda FGX.*(FX)(GX)):(\mathbb{M}\to\mathbb{M}\to\mathbb{M})],\\
[(\lambda FGX.+(FX)(GX)):(\mathbb{M}\to\mathbb{M}\to\mathbb{P})],\\
[\partial:(\mathbb{M}\to\mathbb{P})], [\partial:(\mathbb{P}\to\mathbb{P})], [\partial:(\mathbb{M}\to\mathbb{M})].
\end{array}$$

Example (continued)

- **Question:** Are there non-negative, differentiable functions?
- **Unification Problem:** $G_{(\mathbb{R}\to\mathbb{R}^+)} =^? F_{\mathbb{M}}$
  - guess $G_{(\mathbb{R}\to\mathbb{R}^+)}$ to be $(\lambda X. * (H^1_{(\mathbb{R}\to\mathbb{R})}X)(H^1X))$:

$$F_{\mathbb{M}} =^? (\lambda X. * (H^1_{(\mathbb{R}\to\mathbb{R})}X)(H^1X))$$

  - imitate with $F_{\mathbb{M}}$ as $\lambda X. * (H^2_{\mathbb{M}}X)(H^3_{\mathbb{M}}X)$:

$$H^1_{(\mathbb{R}\to\mathbb{R})}Z^0 =^? H^2_{\mathbb{M}}Z^0 \wedge H^1_{(\mathbb{R}\to\mathbb{R})}Z^0 =^? H^3_{\mathbb{M}}Z^0$$

  - weaken $H^1_{(\mathbb{R}\to\mathbb{R})}$ to $H^4_{\mathbb{M}}$

$$H^4_{\mathbb{M}}Z^0 =^? H^2_{\mathbb{M}}Z^0 \wedge H^4_{\mathbb{M}}Z^0 =^? H^3_{\mathbb{M}}Z^0$$

  - solvable with with $H^4 = H^3 = H^2$
- **Answer:** $F = G = \lambda X_{\mathbb{R}}. * (H^4_{\mathbb{M}}X)(H^4_{\mathbb{M}}X))$ (even degree monomial)

Abductive Reconstruction of Parallelism (ARP)

- Mix Parallelism with HOCU
- Example (Gapping): *John likes Golf and Mary too.*
- Representation $\mathrm{loves}(\mathrm{john}, \mathrm{golf}) \wedge R(\mathrm{mary})$
- Equation $\mathrm{loves}(\mathrm{john}_{\mathrm{john}}, \mathrm{golf}_{\mathrm{golf}}) =^s R^{\neg\mathsf{pe}}_{(\mathbb{W}\mathtt{oman} \to o)}(\mathrm{mary}_{\mathrm{mary}})$
  - R for the missing semantics (of Sort $\mathbb{W}\mathtt{oman} \to o$ and not primary for ellipsis)
- Number Restriction Constraint
  - *Jon* and *golf* might be parallel to *Mary*, but at most one of them can.
  - color variable A: if *Jon* is pe then *golf* isn't, and vice versa
  - Generalizes DSP's Primary Occurrence Restriction (POR)

- Initial Equation: $\mathrm{loves}(\mathrm{john}_{\mathrm{john}},\mathrm{golf}_{\mathrm{golf}})=^{?}R^{\neg\mathrm{pe}}_{(\mathbb{Woman}\to o)}(\mathrm{mary}_{\mathrm{mary}})$
  - imitate $R^{\neg\mathrm{pe}}_{(\mathbb{Woman}\to o)}$ with $\lambda Z.\mathrm{loves}(H_H Z, K_K Z)$
  - H, K new variables of sort $\mathbb{Woman}\to\mathbb{Human}$
- $\mathrm{loves}(\mathrm{john}_{\mathrm{john}},\mathrm{golf}_{\mathrm{golf}})=^{?}\mathrm{loves}(H_H(\mathrm{mary}_{\mathrm{mary}}), K_K\mathrm{mary}_{\mathrm{mary}})$
- $H_H\mathrm{mary}_{\mathrm{mary}}=^{?}\mathrm{john}_{\mathrm{john}} \wedge K_K\mathrm{mary}_{\mathrm{mary}}=^{?}\mathrm{golf}_{\mathrm{golf}}$
- Two possible continuations:

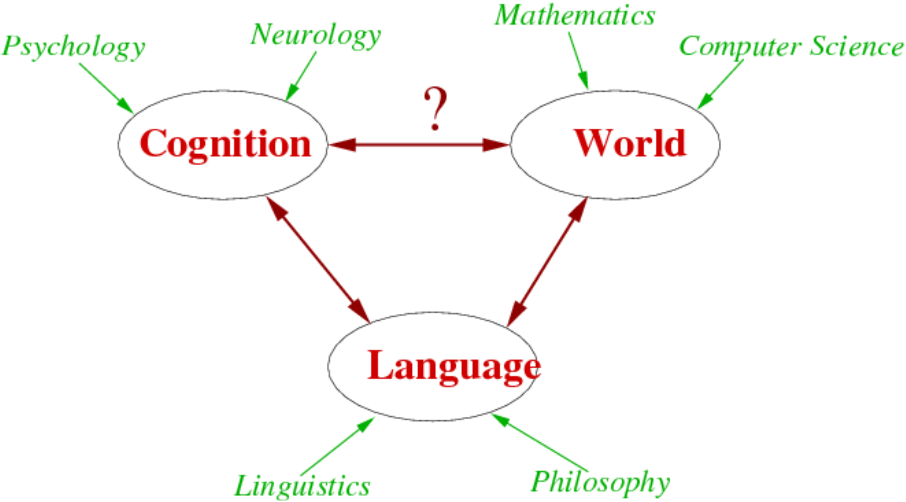
- project $H = \lambda Z.Z$ (so $A=^{?}\mathrm{pe}$)
- imitate $K = \lambda Z.\mathrm{golf}_{\mathrm{golf}}$
- then $\boxed{\begin{array}{l}\mathrm{mary}_{\mathrm{mary}}=^{?}\mathrm{john}_{\mathrm{john}}\\ \mathrm{golf}_{\mathrm{golf}}=^{?}\mathrm{golf}_{\mathrm{golf}}\end{array}}$
- *Mary likes Golf* (*preferred*)

- project $K = \lambda Z.Z$ (so $\neg A=^{?}\mathrm{pe}$)
- imitate $H = \lambda Z.\mathrm{john}_{\mathrm{john}}$
- then $\boxed{\begin{array}{l}\mathrm{john}_{\mathrm{john}}=^{?}\mathrm{john}_{\mathrm{john}}\\ \mathrm{mary}_{\mathrm{mary}}=^{?}\mathrm{golf}_{\mathrm{golf}}\end{array}}$
- John *likes Mary*

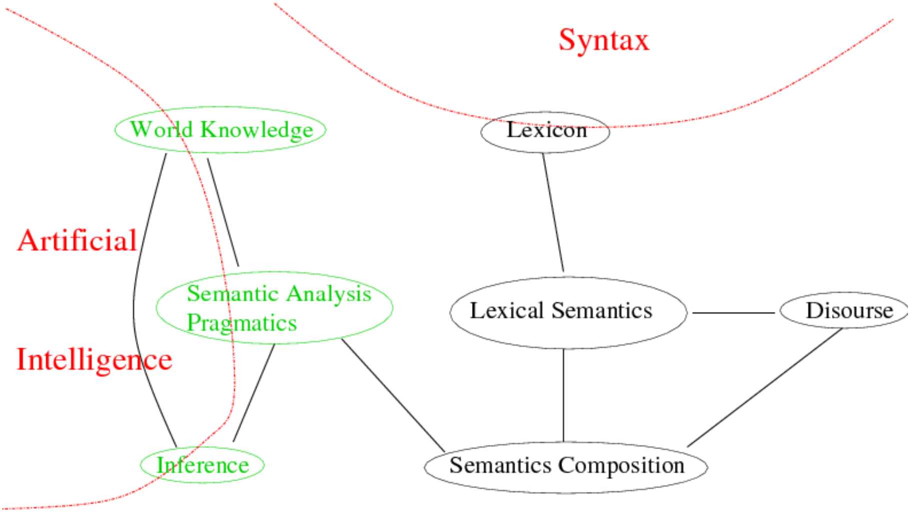
Chapter 14
Conclusion

14.1 A Recap in Diagrams

NL Semantics as an Intersective Discipline



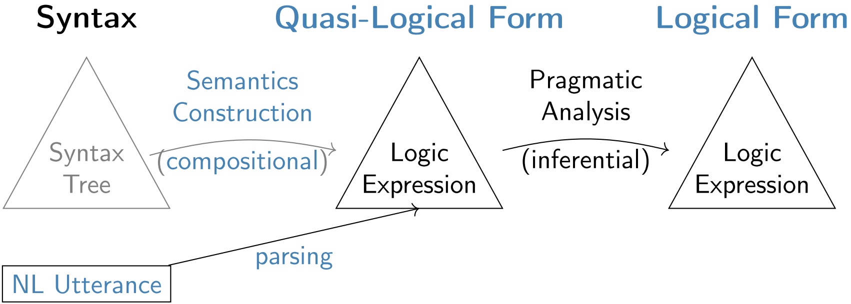
A landscape of formal semantics



Modeling Natural Language Semantics

- **Problem:** Find formal (logic) system for the meaning of natural language.
- History of ideas
  - Propositional logic [ancient Greeks like Aristotle]
    - *Every human is mortal*
  - First-Order Predicate logic [Frege $\leq$ 1900]
    - *I believe, that my audience already knows this.*
  - Modal logic [Lewis18, Kripke65]
    - *A man sleeps. He snores.* $((\exists X.\mathrm{man}(X) \wedge \mathrm{sleeps}(X))) \wedge \mathrm{snores}(X)$
  - Various dynamic approaches (e.g. DRT, DPL)
    - *Most men wear black*
  - Higher-order Logic, e.g. generalized quantifiers
  - ...

A Semantic Processing Pipeline based on LF



Syntax

Quasi-Logical Form

Logical Form

Syntax Tree

Semantics Construction (compositional)

Logic Expression

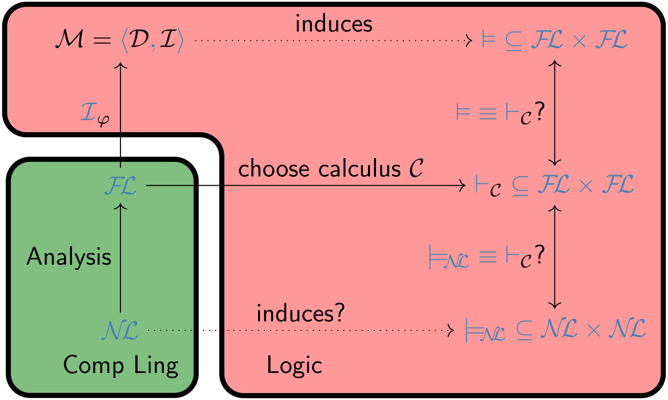
Pragmatic Analysis (inferential)

Logic Expression

NL Utterance

parsing

Natural Language Semantics?



$$\begin{tikzcd}
\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle \arrow[rr, dotted, "\text{induces}"] & & \models \subseteq \mathcal{FL} \times \mathcal{FL} \arrow[d, leftrightarrow, "\models \equiv \vdash_{\mathcal{C}}?"'] \\
\mathcal{FL} \arrow[u, "\mathcal{I}_\varphi"] \arrow[rr, "\text{choose calculus } \mathcal{C}"] & & \vdash_{\mathcal{C}} \subseteq \mathcal{FL} \times \mathcal{FL} \arrow[d, leftrightarrow, "\models_{\mathcal{NL}} \equiv \vdash_{\mathcal{C}}?"'] \\
\mathcal{NL} \arrow[u, "\text{Analysis}"] \arrow[rr, dotted, "\text{induces?}"] & & \models_{\mathcal{NL}} \subseteq \mathcal{NL} \times \mathcal{NL}
\end{tikzcd}$$

Comp Ling

Logic

14.2 Where to From Here

Where to from here?

- We can continue the exploration of semantics in two different ways:
  - Look around for additional logical/formal systems and see how they can be applied to various linguistic problems. (the logician's approach)
  - Look around for additional linguistic forms and wonder about their truth conditions, their logical forms, and how to represent them. (the linguist's approach)
- Here are some possibilities...

Semantics of Plurals

1. *The dogs were barking.*
2. *Fido and Chester were barking.* (What kind of an object do the subject NPs denote?)
3. *Fido and Chester were barking. They were hungry.*
4. *Jane and George came to see me. She was upset.* (Sometimes we need to look inside a plural!)
5. *Jane and George have two children.* (Each? Or together?)
6. *Jane and George got married.* (To each other? Or to other people?)
7. *Jane and George met.* (The predicate makes a difference to how we interpret the plural)

Reciprocals

- What's required to make these true?
  1. *The men all shook hands with one another.*
  2. *The boys are all sitting next to one another on the fence.*
  3. *The students all learn from each other.*

Presuppositional expressions

- What are presuppositions?
- What expressions give rise to presuppositions?
- Are all apparent presuppositions really the same thing?
  1. *The window in that office is open.*
  2. *The window in that office isn't open.*
  3. *George knows that Jane is in town.*
  4. *George doesn't know that Jane is in town.*
  5. *It was / wasn't George who upset Jane.*
  6. *Jane stopped / didn't stop laughing.*
  7. *George is / isn't late.*

Presupposition projection

1. *George doesn't know that Jane is in town.*
2. *Either Jane isn't in town or George doesn't know that she is.*
3. *If Jane is in town, then George doesn't know that she is.*
4. *Henry believes that George knows that Jane is in town.*

Conditionals

- What are the truth conditions of conditionals?
  1. *If Jane goes to the game, George will go.*
     - Intuitively, not made true by falsity of the antecedent or truth of consequent independent of antecedent.
  2. *If John is late, he must have missed the bus.*
- Generally agreed that conditionals are modal in nature. Note presence of modal in consequent of each conditional above.

Counterfactual conditionals

- And what about these??
  1. *If kangaroos didn't have tails, they'd topple over.* (David Lewis)
  2. *If Woodward and Bernstein hadn't got on the Watergate trail, Nixon might never have been caught.*
  3. *If Woodward and Bernstein hadn't got on the Watergate trail, Nixon would have been caught by someone else.*
- Counterfactuals undoubtedly modal, as their evaluation depends on which alternative world you put yourself in.

Before and after

- These seem easy. But modality creeps in again...
  1. *Jane gave up linguistics after she finished her dissertation.* (Did she finish?)
  2. *Jane gave up linguistics before she finished her dissertation.* (Did she finish? Did she start?)

References I

[Ari10] Mira Ariel. *Defining Pragmatics*. Research Surveys in Linguistics. Cambridge University Press, 2010.

[BB05] Patrick Blackburn and Johan Bos. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI, 2005.

[Ben91] Johan van Benthem. *Language in Action, Categories, Lambdas and Dynamic Logic*. Vol. 130. Studies in Logic and Foundation of Mathematics. North Holland, 1991.

[Bir13] Betty J. Birner. *Introduction to Pragmatics*. Wiley-Blackwell, 2013.

[Bla+01] Patrick Blackburn et al. "Inference and Computational Semantics". In: *Computing Meaning (Volume 2)*. Ed. by Harry Bunt et al. Kluwer Academic Publishers, 2001, pp. 11–28.

[BRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. New York, NY, USA: Cambridge University Press, 2001. ISBN: 0-521-80200-8.

[Cho65] Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965.

References II

[Chu40] Alonzo Church. "A Formulation of the Simple Theory of Types". In: *Journal of Symbolic Logic* 5 (1940), pp. 56–68.

[CKG09] Ronnie Cann, Ruth Kempson, and Eleni Gregoromichelaki. *Semantics – An Introduction to Meaning in Language*. Cambridge University Press, 2009. ISBN: 0521819628.

[Cre82] M. J. Cresswell. "The Autonomy of Semantics". In: *Processes, Beliefs, and Questions: Essays on Formal Semantics of Natural Language and Natural Language Processing*. Ed. by Stanley Peters and Esa Saarinen. Springer, 1982, pp. 69–86. DOI: 10.1007/978-94-015-7668-0_2.

[Cru11] Alan Cruse. *Meaning in Language: An Introduction to Semantics and Pragmatics*. Oxford Textbooks in Linguistics. 2011.

[Dav67a] Donald Davidson. "The logical form of action sentences". In: *The logic of decision and action*. Ed. by N. Rescher. Pittsburgh: Pittsburgh University Press, 1967, pp. 81–95.

[Dav67b] Donald Davidson. "Truth and Meaning". In: *Synthese* 17 (1967).

References III

[de 95] Manuel de Vega. “Backward updating of mental models during continuous reading of narratives”. In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 21 (1995), pp. 373–385.

[EU10] Jan van Eijck and Christina Unger. *Computational Semantics with Functional Programming*. Cambridge University Press, 2010.

[Fre92] Gottlob Frege. “Über Sinn und Bedeutung”. In: *Zeitschrift für Philosophie und philosophische Kritik* 100 (1892), pp. 25–50.

[GML87] A. M. Glenberg, M. Meyer, and K. Lindem. “Mental models contribute to foregrounding during text comprehension”. In: *Journal of Memory and Language* 26 (1987), pp. 69–83.

[Göd32] Kurt Gödel. “Zum Intuitionistischen Aussagenkalkül”. In: *Anzeiger der Akademie der Wissenschaften in Wien* 69 (1932), pp. 65–66.

[Gol81] Warren D. Goldfarb. “The Undecidability of the Second-Order Unification Problem”. In: *Theoretical Computer Science* 13 (1981), pp. 225–230.

References IV

[Har84] D. Harel. "Dynamic Logic". In: *Handbook of Philosophical Logic*. Ed. by D. Gabbay and F. Günthner. Vol. 2. Reidel, Dordrecht, 1984, pp. 497–604.

[HC84] G. E. Hughes and M. M. Cresswell. *A companion to Modal Logic*. University Paperbacks. Methuen, 1984.

[HHS07] James R. Hurford, Brendan Heasley, and Michael B. Smith. *Semantics: A coursebook*. 2nd. Cambridge University Press, 2007.

[HK00] Dieter Hutter and Michael Kohlhase. "Managing Structural Information by Higher-Order Colored Unification". In: *Journal of Automated Reasoning* 25.2 (2000), pp. 123–164. URL: `https://kwarc.info/kohlhase/papers/jar00.pdf`.

[HM95] Furio Honsell and Marino Miculan. "A natural deduction approach to dynamic logic". In: *Types for Proofs and Programs TYPES '95*. Ed. by Stefano Berardi and Mario Coppo. 1995, pp. 165–182. ISBN: 978-3-540-61780-8. DOI: `10.1007/3-540-61780-9_69`.

References V

[Hue76] Gérard P. Huet. “Résolution d’Équations dans des Langages d’ordre 1,2,...,w.”. Thèse d‘État. Unif-bib: Université de Paris VII, 1976.

[Isr93] David J. Israel. “The Very Idea of Dynamic Semantics”. In: *Proceedings of the Ninth Amsterdam Colloquium*. 1993. URL: `https://arxiv.org/pdf/cmp-lg/9406026.pdf`.

[Jac83] Ray Jackendoff. *Semantics and Cognition*. MIT Press, 1983.

[JL83] P. N. Johnson-Laird. *Mental Models*. Cambridge University Press, 1983.

[JLB91] P. N. Johnson-Laird and Ruth M. J. Byrne. *Deduction*. Lawrence Erlbaum Associates Publishers, 1991.

[Kea11] Kate Kearns. *Semantics*. 2nd. Palgrave Macmillan, 2011.

[Kon04] Karsten Konrad. *Model Generation for Natural Language Interpretation and Analysis*. Vol. 2953. LNCS. Springer, 2004. ISBN: 3-540-21069-5. DOI: `10.1007/b95744`.

[Kri63] Saul Kripke. “Semantical Considerations on Modal Logic”. In: *Acta Philosophica Fennica* (1963), pp. 83–94.

References VI

[Lew18] Clarence Irving Lewis. *A Survey of Symbolic Logic*. University of California Press, 1918. URL: `http://hdl.handle.net/2027/hvd.32044014355028`.

[Mat70] Ju. V. Matijasevič. “Enumerable sets are diophantine”. In: *Soviet Math. Doklady* 11 (1970), pp. 354–358.

[MBV95] Reinhard Muskens, Johan van Benthem, and Albert Visser. “Dynamics”. In: ed. by Johan van Benthem and Ter Meulen. Elsevier Science B.V., 1995.

[MR98] C. Monz and M. de Rijke. “A Resolution Calculus for Dynamic Semantics”. In: *Logics in Artificial Intelligence. European Workshop JELIA ’98*. LNAI 1489. Springer Verlag, 1998.

[Nor+18a] Emily Nordmann et al. *Lecture capture: Practical recommendations for students and lecturers*. 2018. URL: `https://osf.io/huydx/download`.

[Nor+18b] Emily Nordmann et al. *Vorlesungsaufzeichnungen nutzen: Eine Anleitung für Studierende*. 2018. URL: `https://osf.io/e6r7a/download`.

References VII

[Ohl88] Hans Jürgen Ohlbach. "A Resolution Calculus for Modal Logics". PhD thesis. Universität Kaiserslautern, 1988.

[Par90] Terence Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics*. Vol. 19. Current Studies in Linguistics. MIT Press, 1990.

[Por04] Paul Portner. *What is Meaning? Fundamentals of Formal Semantics*. Blackwell, 2004.

[Pra76] V. Pratt. "Semantical considerations of Floyd-Hoare logic". In: *Proceedings of the 17th Symposium on Foundations of Computer Science*. 1976, pp. 109–121.

[Ran17] Aarne Ranta. *Automatic Translation for Consumers and Producers*. Presentation given at the Chalmers Initiative Seminar. 2017. URL: `https://www.grammaticalframework.org/~aarne/mt-digitalization-2017.pdf`.

[RG94] Uwe Reyle and Dov M. Gabbay. "Direct Deductive computation on Discourse Representation Structures". In: *Linguistics & Philosophy* 17 (1994), pp. 343–390.

References VIII

[Rie10] Nick Riemer. *Introducing Semantics*. Cambridge Introductions to Language and Linguistics. Cambridge University Press, 2010.

[Rus03] Bertrand Russell. *The Principles of Mathematics*. Cambridge University Press, 1903.

[Rus91] Stuart J. Russell. "An Architecture for Bounded Rationality". In: *SIGART Bulletin* 2.4 (1991), pp. 146–150.

[Sae03] John I. Saeed. *Semantics*. 2nd. Blackwell, 2003.

[Sau93] Werner Saurer. "A Natural Deduction System for Discourse Representation Theory". In: *Journal of Philosophical Logic* 22 (1993).

[Sch20] Jan Frederik Schaefer. "Prototyping NLU Pipelines – A Type-Theoretical Framework". Master's Thesis. Informatik, FAU Erlangen-Nürnberg, 2020. URL: https://gl.kwarc.info/supervision/MSc-archive/blob/master/2020/Schaefer_Jan_Frederik.pdf.

References IX

[Sin94] M. Singer. “Discourse Inference Processes”. In: *Handbook of Psycholinguistics*. Ed. by M. A. Gernsbacher. Academic Press, 1994, pp. 479–515.

[Sta14] Robert Stalnaker. *Context*. Oxford University Press, 2014.

[Ven57] Zeno Vendler. “Verbs and times”. In: *Philosophical Review* 56 (1957), pp. 143–160.

[ZR98] R. A. Zwaan and G. A. Radvansky. “Situation models in language comprehension and memory”. In: *Psychological Bulletin* 123 (1998), pp. 162–185.

[ZS13] Thomas Ede Zimmermann and Wolfgang Sternefeld. *Introduction to Semantics*. de Gruyter Mouton, 2013.