VARIABLES AS FUNCTIONS:
EFFICIENT REFERENCE TO DEPENDENT ENTITIES

Lenhart Schubert
University of Rochester

Thanks: Ken Shan, NSF grants IIS–0082928, IIS–0328849
FUNCTIONAL REFERENCE

(1) All of the graduates received a job offer (at the job fair), and all of them accepted their offer.

(2) If all of the graduates received a job offer then all of them accepted their offer.

(cf. donkey sentences)
LOGICAL FORMS FOR FUNCTIONAL REFERENCE

(1) All of the graduates received a job offer (at the job fair), and all of them accepted their offer.

a. faulty LF for (1):
\((\forall x) [graduate(x) \rightarrow (\exists y) job-offer(y) \& receive(x,y)]\)
& \((\forall x) [graduate(x) \rightarrow accept(x,y)]\)

b. verbose LF for (1):
\((\forall x) [graduate(x) \rightarrow (\exists y) job-offer(y) \& receive(x,y)]\)
& \((\forall x) [[graduate(x) \& (\exists y) job-offer(y) \& receive(x,y)] \rightarrow accept(x,y)]\)

c. functional LF for (1):
\((\forall x) [graduate(x) \rightarrow (\exists y) job-offer(y) \& receive(x,y)]\)
& \((\forall x) [graduate(x) \rightarrow accept(x,y(x))]\)
\exists -variables as functions: Intuitive idea

e.g., Everyone has a mother & no-one has a clone:

\[ (\varphi) \quad \forall x. \exists y \text{ mother-of}(x,y) \land \neg \exists z \text{ clone-of}(x,z) \]

Satisfaction set for \( \varphi \), relative to model \( M = (D,I) \)?

\[ \llbracket \varphi \rrbracket_M \approx \{ <U,V> | U, V \text{ are the same except that } V(y) \]

is a function \( D \rightarrow D \) picking out "the"

mother of any given \( d \) in \( D \} \]

(We don’t need such a function for \( z \! \)

Negated formulas are "static")

We could then use

\( y(x) \) for "the mother of \( x \)";
\( y(y(x)) \) for the grandmother of \( x \);
\( y(Oedipus) = Jocasta; \) etc.
"Functional DPL": Remarks on syntax

- Assume occurrence of $\exists y$ is unique:
  We want permanent use of the implicit functions.

  $$\forall x. \exists y \text{ mother-of}(x, y) \land \exists y \text{ father-of}(x, y)$$

  would yield only one function (for the father)

- Should we insist on functions getting the "right" number of arguments?

  We could, using the notion of "defining context".
  But instead we allow "abuses" like

  $$y(\text{Oedipus})(\text{Jocasta}), \text{ Oedipus}(y),$$

  and ensure predications involving such terms are false.
**"Functional DPL": Semantics**

- Generalized variable assignments (gva’s) $U$, $V$, $W$, ...

  of type $\text{Var} \rightarrow \mathcal{F}$

  where $\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots$

  $\mathcal{F}_n = D \rightarrow D = D \rightarrow (D \rightarrow (\ldots (D \rightarrow D)\ldots))$

- $\llbracket \exists y \varphi \rrbracket_M = \{<U,V> \mid \text{for some } d \in D, \ <U_{y:d},V> \in \llbracket \varphi \rrbracket_M\}$

- $\llbracket \forall x \varphi \rrbracket_M = \{<U,V> \mid \text{for all } d \in D, \ <U_{x:d},V'_{x:d}> \in \llbracket \varphi \rrbracket_M, \text{ where for all var’s } y, V'(y) = V(y) = U(y) \}$

  if for all $<W,W'> \in \llbracket \varphi \rrbracket_M$,

  $W'(y) = W(y)$,

  and otherwise $V'(y) = V(y)(d)$

"screens out" all var’s $y$ that are not $\mathcal{F}$-quantified in $\varphi$

(or are $\mathcal{F}$-quantified in a static subformula, like the earlier "clone-of" subformula)
Semantics (cont’d)

- Predication, negation, conjunction are as in DPL
  but (crucially) disjunction & conditional are dynamic:

• $\langle \phi \lor \psi \rangle_M = \langle \phi \rangle_M \cup \langle \psi \rangle_M$

• $\langle \phi \rightarrow \psi \rangle_M = \{<U,V> | \text{either } V=U \text{ & for no gva } U',
  \text{("weak" conditional) } <U,U'> \in \langle \phi \rangle_M, \text{ or for some gva } U',
  <U,U'> \in \langle \phi \rangle_M \land <U',V> \in \langle \psi \rangle_M \}$

E.g., John has a dog or a cat;
he keeps it in the house.

E.g., If John is lucky, he’ll get a fax from Mary;
(?) it will contain a job offer.
Semantics (concluded)

- **Truth:** \( M, U \models \varphi \iff \text{for some gva } V, \langle U, V \rangle \in \llbracket \varphi \rrbracket_M \)

- **Entailment:**
  \[ \varphi \models \psi \iff \text{for all models } M \text{ and all } U, V, \]
  \[ \text{if } \langle U, V \rangle \in \llbracket \varphi \rrbracket_M \text{ then for some gva's } \]
  \[ V', W, \langle U, V' \rangle \in \llbracket \varphi \rrbracket_M \text{ & } \langle V', W \rangle \in \llbracket \psi \rrbracket_M \]

  This aligns with \( \models \) with \( \rightarrow : \)
  \[ \varphi \models \psi \iff \models \varphi \rightarrow \psi \]

  Also we have the "detachment"
  \[ [\exists x \ P(x) \rightarrow Q(x)], J y \ P(y) \models Q(x) \]

- **The semantics can be generalized to restricted quantifiers**
  \( (\forall x: \varphi) \psi, (\text{Most } x: \varphi) \psi, \) etc.
Remarks on mapping anaphors to functional expressions

(3) Every student x wrote a paper y;

No student z who took the exam handed in their paper on time

y(z)

(4) Every student x that didn’t take the exam wrote a paper y;

No student z handed in their paper on time.*

y(z)? NO!

* Example due to Ken Shan
Remarks on mapping anaphors
to functional expressions (cont’d)

Suppose the defining context for \( y \) is
\[
(\forall x: C(x)) \exists y \ldots
\]
and we’re considering a preliminary LF of form
\[
(\forall z: C'(z)) \phi [\text{the } P].
\]

Does substitution of \( y(z) \) for "the P" yield a potential reading?

A necessary condition (for the case of atomic \( C, C' \)):
\[
I(C') \subseteq I(C) \text{ in the intended model } M = (D,I)
\]

(along with other "standard" constraints).
Bridging anaphora

(5) Prior knowledge: Every house has a front door

\( (\forall x) \ [\text{house}(x) \rightarrow \exists y. \text{door}(y) \& \text{part-of}(y,x) \& \text{at-front-of}(y,x)] \)

(6) New facts:
   a. Cora walked up to the house \( z \);
   b. She knocked on the door.

   \( \text{knock-on}(\text{Cora}, y(z)) \)

Advantages:

- Simplicity

- Avoids uniqueness presumption

  (Houses can have multiple doors, even in front!)
Frames, scripts, generic sentences

(5) Again consider: Every house has a front door

\( \forall x \) [house(x) \rightarrow \exists y. \text{door(y)} \& \text{part-of(y,x)} \& \text{at-front-of(y,x)}] 

The "creation" of a function \( y \) can be viewed as the creation of a frame slot -- directly via NLP!

(7) Similarly for events:
When someone eats [e] at a restaurant,
they enter [e1], find [e2] a table to sit at,
wait [e4] for the server, select [e4] a meal, ...

Both subevents and roles (table, server, etc.)
could again be created directly via NLP ==> scripts.

A remaining problem: For non-universal generic quantifiers, the functions created may have some "arbitrary" values, and this can lead to faulty LFs for sentences with functional reference.

A possible solution: partial functions!
Conclusions & further work

- By letting $\exists$-variables acquire functional values in $\forall$-contexts, we can easily represent functional reference.

- The resulting "functional DPL" differs in some minor respects from DPL: among the logical operators, only negation is uniformly static.

- Further work:
  - exploration of the logic
  - translation to FOL
  - more detailed study of NL $\rightarrow$ LF mapping for functional anaphora
  - develop a partial-function variant; hence deal with dependencies on generic quantifiers other than $\forall$. 