Parallel design in DPL

- Dynamic systems like DPL let indefinites bind pronouns across sentences:

  S_1: A^f farmer owns a^d donkey. S_2: He$_f$ doesn't beat it$_d$.

  $S_1 + S_2 \rightsquigarrow$ $\mathbf{[f];\ farmer(f);\ [d];\ donkey(d);\ owns(f,d);\ \neg beats(f,d)}$

- Donkey anaphora (Geach 1962) is formally identical in DPL:

  S_3: Every^f farmer who owns a^d donkey beats it$_d$.

  $\rightsquigarrow$ "There is no value for f that satisfies $S_1 + S_2$"

  $\rightsquigarrow \neg \left(\mathbf{[f];\ farmer(f);\ [d];\ donkey(d);\ owns(f,d);\ \neg beats(f,d)} \right)$

DPL refresher

- A DPL formula determines an **update** relation $\longrightarrow$ between an input variable assignment and an output assignment.

For any variables ν, ν', predicate P, formulas ϕ, ψ and assignments g, h,

Random Assignment	$g \xrightarrow{[\nu]} g[\nu/\alpha]$	for any individual α
Predicate	$g \circlearrowleft P(\nu)$	iff P holds of $g(\nu)$
Literal	$g \circlearrowleft P(\nu, \nu')$	iff P holds of $\left\langle g(\nu), g(\nu') \right\rangle$
Conjunction	$g \xrightarrow{\phi;\psi} h$	iff $\exists$ k s.t. $g \xrightarrow{\phi} k \xrightarrow{\psi} h$
Negation	$g \circlearrowleft \neg\phi$	iff $g \xrightarrow{\phi}$ ✖

What DPL doesn't do

- Plurals, including **discourse plurals**:
  - Most students were late. They are not as punctual as professors.
  - Most students were late. They got lower grades than the rest.
- Quantificational Subordination:
  - Every student wrote a paper. Most turned it in on time.
- Paycheck pronouns:
  - Most employees deposited the bonus they got .
  - Some cashed it instead.

Intuition behind the new logic

Most^s students in my class...

... wrote a^p final paper.

Each^s submitted it$_p$ on time.

S_1: There's a^s student in my class.

S_2: She$_s$ wrote a^p final paper.

S_3: She$_s$ submitted it$_p$ on time.

- We can paraphrase the subordination in terms of S_1, S_2, and S_3:
  - "Most s values satisfying S_1 also satisfy $S_1 + S_2$."
  - "All s values satisfying $S_1 + S_2$ also satisfy $S_1 + S_2 + S_3$."
- Using this method, the anaphora in the subordination discourse is formally identical to that in the simpler discourse $S_1 + S_2 + S_3$.

Implementing the intuition: update variables

- First, we introduce new **update variables** to store full DPL formula meanings (i.e., updates): $A_1, A_2, \ldots, Z_1, Z_2, \ldots$
- Quantification over an *individual* variable like s introduces the corresponding *update* variables S_1 and S_2.
- The nuclear scope is anaphorically linked to the restrictor.
- Finally, a subordinated restrictor is entirely anaphoric (though we still relabel it for consistency):

Translating subordination

Most [students in my class $]^{S_1}$ [wrote a^p final paper $]^{S_2}_{S_1}$

Each [$]^{S_3}_{S_2}$ [submitted it$_p$ on time $]^{S_4}_{S_3}$

Implementing the intuition: update assignments

- Update variables are introduced in the logic via **update assignments**:
  - $(\mathbf{S_1 : student(s); in\text{-}my\text{-}class(s)})$
- Previously introduced update variables may function as clauses to define anaphoric links:
  - $(\mathbf{S_2 : S_1; [p]; final\text{-}paper(p); wrote(s,p)}) =$ $(\mathbf{S_2 : student(s); in\text{-}my\text{-}class(s);\ [p]; final\text{-}paper(p); wrote(s,p)})$
  - $(\mathbf{S_4 : S_3; submitted\text{-}on\text{-}time(s,p)}) =$ $(\mathbf{S_4 : student(s); in\text{-}my\text{-}class(s); \ldots; submitted\text{-}on\text{-}time(s,p)})$
- Notice that we made the choice to essentially store open formulas as updates – i.e., there is no "[*s*]" term. We return to this point below.

Translating subordination

Most [students in my class $]^{S_1}$ [wrote a^p final paper $]^{S_2}_{S_1}$

Each [$]^{S_2}_{S_3}$ [submitted it$_p$ on time $]^{S_4}_{S_3}$

Implementing the intuition: complex terms

- Finally, we need **complex variable terms** to extract the set of all possible outputs for an individual variable within an update:
  - $\mathbf{S_1.s}$ (students in my class)
  - $\mathbf{S_2.s}$ (members of $\mathbf{S_1.s}$ who wrote a paper)
  - $\mathbf{S_3.s}$ (members of $\mathbf{S_2.s}$ who submitted on time)
  - $\mathbf{S_2.p}$ (papers written by a member of $\mathbf{S_2.s}$), etc.
- With appropriate set-theoretic definitions of quantifiers, we can now complete the translation of our subordination example.
- Complex terms are also the appropriate values for discourse plurals: They$_{S_2.p}$ are collecting dust in a pile on my desk.

Translating subordination

Most students in my class wrote a final paper. Each submitted it on time.
$(S_1 : \mathit{student}(s); \mathit{in\text{-}my\text{-}class}(s)); (S_2 : S_1; [p]; \mathit{final\text{-}paper}(p); \mathit{wrote}(s, p));$
$(S_3 : S_2; \mathit{submitted\text{-}on\text{-}time}(s, p)); \ \mathit{MOST}(S_1.s, S_2.s); \mathit{ALL}(S_2.s, S_3.s)$

Values of complex terms

- One choice of value for the complex term "$S_2.p$", given an assignment g, would be the set of output values for p after S_2 applies to g:
$$\left\{ h(p) : g \xrightarrow{\textbf{student(s); in-my-class(s); [p]; final-paper(p); wrote(s, p)}} h \right\}$$
- Recall, though, that we chose to leave s free in S_2. Thus, this set will contain only the paper written by the single student $g(s)$.
- What we want, rather, is the set of papers written by all students.
- Thus, we assume an implicit random assignment complex term values:
$$\left\{ h(p) : g \xrightarrow{\textbf{[s]; student(s); in-my-class(s); \ldots}} h \right\}$$
- The choice to introduce random assignment here rather than in the value for S_2 seems arbitrary. However, the analysis of strong donkey pronouns below is greatly facilitated by this move.

DUAL semantics: preliminaries

- DUAL semantics uses two assignments, an **individual assignment** g like DPL and an **update assignment** G, linking update variables to DPL-style relations: g G
- To save space, we will play a little fast and loose with the distinction between a logical formula ϕ and the DPL *update* g $\xrightarrow{\phi}$ h corresponding to this formula.
  - I.e, for formula ϕ and update variable $\widehat{\nu}_n$, we will write $G[\widehat{\nu}_n/\phi]$ instead of $G\left[\widehat{\nu}_n/ g \xrightarrow{\phi} h\right]$

DUAL semantics

For any variables $\nu, \nu', \hat{\nu}_n$, predicate P, variable terms $\tau_1, \ldots, \tau_m$, formulas ϕ,ψ, and DUAL assignments $\boxed{g \; G}$ and $\boxed{h \; H}$:

Individual Assignment

$\boxed{g \; G} \xrightarrow{[\nu]} \boxed{g[\nu/\alpha] \; G}$ for any individual α

Predicate Literal

$\boxed{g \; G} \circlearrowright^{P(\tau_1, \ldots, \tau_m)}$ iff P holds of $\langle [\![\tau_1]\!]_g, \ldots, [\![\tau_m]\!]_g \rangle$, where

$[\![\nu]\!]_g = \{ g(\nu) \}$ and $[\![\hat{\nu}_n.\nu']\!]_g = \left\{ h(\nu') : g \xrightarrow{[\nu];\; G(\hat{\nu}_n)} h \right\}$

Conjunction

$\boxed{g \; G} \xrightarrow{\phi;\psi} \boxed{h \; H}$ iff $\exists$ $\boxed{k \; K}$ s.t.

$\boxed{g \; G} \xrightarrow{\phi} \boxed{k \; K} \xrightarrow{\psi} \boxed{h \; H}$

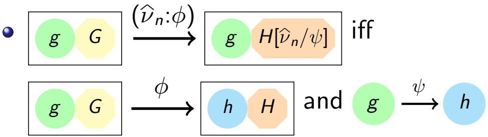
Update Literal

$\boxed{g \; G[\hat{\nu}_n/\phi]} \xrightarrow{\hat{\nu}_n} \boxed{h \; G[\hat{\nu}_n/\phi]}$ iff $g \xrightarrow{\phi} h$

Update Assignment

$\boxed{g \; G} \xrightarrow{(\hat{\nu}_n:\phi)} \boxed{g \; G[\hat{\nu}_n/\phi]}$

1. The system presented on the previous page does not handle embedded update assignments, such as $(A_1 : (B_1 : \ldots)\ldots)$.
Full definition:



2. DUAL Negation can be defined in terms of update assignments:

$$\sim\phi := \sim^m\phi := ((N_m : \phi); |N_m.n|{=}0)$$

where m is the number of previous occurrences of negation.

- Since "$N_m.n$" involves the random assignment "$[n]$", its cardinality is only zero when ϕ never holds / is false.
- Each occurrence of negation technically has a unique superscript (usually suppressed) in order to distinguish embedded negations.

Weak donkey anaphora

- Having seen how DUAL handles discourse plurals and quantificational subordination, let's return to donkey anaphora:
  Every [student who owns a^c car]^{S_1} [drove it$_c$ to the party]$^{S_2}_{S_1}$.
  - $\mathbf{(S_1 : student(s); [c]; car(c); owns(s, c));}$
    $\mathbf{(S_2 : S_1; drove(s, c)); EVERY(S_1.s,\ S_2.s)}$
  - $[\![S_1.s]\!]_g = \{h(s) : g \xrightarrow{[s];\ student(s);\ [c];\ car(c);\ owns(s,\ c)} h\}$
  - $[\![S_2.s]\!]_g = \{h(s) : g \xrightarrow{[s];\ student(s);\ [c];\ car(c);\ owns(s,\ c);\ drove(s,\ c)} h\}$
  - "Every student who owns a car (is a student who) drove a car they own to the party."
- This derives the weak reading of the donkey pronoun. The analysis of strong donkey pronouns incorporates elements from two other constructions. . .

Telescoping

- Telescoping (Roberts 1987) is like quantificational subordination without the second quantifier:
  - Each chess set includes $\boxed{\text{a}^p \text{ spare pawn}}$. $\boxed{\text{It}_p}$ is taped to the bottom.
- The first sentence is a standard quantified sentence:
  - Each [chess set]^{S_1} [includes a^p spare pawn]$^{S_2}_{S_1}$.
  - **$(S_1 : set(s)); (S_2 : S_1; [p]; pawn(p); includes(s, p)); EVERY(S_1.s, S_2.s)$**
    "Every set is a set with a pawn."
- The second, though, uses a covert universal quantifier $\forall$ whose restrictor is anaphoric to the first sentence:
  - $\forall$ [ ]$^{S_3}_{S_2}$ [It$_p$ is taped to the bottom]$^{S_4}_{S_3}$.
  - **$(S_3 : S_2); (S_4 : S_3; taped\text{-}to\text{-}bottom(p, s)); EVERY(S_3.s, S_4.s)$**
    "Every set with a pawn is a set with a pawn taped to the bottom."

Paycheck pronouns

- Paycheck pronouns (Karttunen 1969) involve reference to an item never mentioned before:
  - Most employees deposited their paycheck.
    But some cashed it (and went on a spree).
- Most employees^{E_1} [deposited [their$_e$ paycheck]^{P_1}]$^{E_2}_{E_1}$.
  - $\mathbf{(E_1 : emp(e)); (E_2 : E_1; (P_1 : pc\text{-}of(p, e)); deposited(e, P_1.p)); MOST(E_1.e, E_2.e)}$ [ignoring the definite's presuppositions]
  - $[\![P_1.p]\!]_g = \{h(p) : g \xrightarrow{[p];pc\text{-}of(p,e)} h\}$
    (Note the free occurrence of e – uses the local value g(e))
  - Hence: "Most employees are employees e who deposited e's paycheck."
- Some [ ]$^{E_3}_{E_1}$ [cashed it$_{P_1.p}$]$^{E_4}_{E_3}$
  - $\mathbf{(E_3 : E_1); (E_4 : E_3; cashed(e, P_1.p)); SOME(E_3.e, E_4.e)}$
  - Here, e is bound to a different quantifier, yielding a different $P_1.p$.
  - Hence: "Some employees are employees e who cashed e's paycheck."

Strong donkey anaphora analysis

- Therefore DUAL assumes a covert quantifier, just as in telescoping:
  - Every $\left[\text{farmer who owns a}^d \text{ donkey}\right]^{F_1} \left[\forall \ [F_1]^{D_1} \ [\text{ beats it}_d]^{D_2}_{D_1}\right]^{F_1}_{F_2}$
  - $\mathbf{(F_1 : farmer(f); [d]; donkey(d); owns(f, d));}$
    $\mathbf{(F_2 : F_1; (D_1 : F_1); (D_2 : D_1; beats(f, d)); EVERY(D_1.d, D_2.d));}$
    $\mathbf{EVERY(F_1.f, F_2.f)}$
  - "Every farmer who owns a donkey is a farmer f who owns a donkey and beats every donkey f owns."
- Although D_1 and F_1 are identical in value, crucially $D_1.d$ and $F_1.d$ differ:
  - $[\![F_1.d]\!]_g = \{h(s) : g \xrightarrow{[f];\ farmer(f);\ [d];\ donkey(d);\ owns(f,\ d)} h\}$
  - $[\![D_1.d]\!]_g = \{h(s) : g \xrightarrow{[d];\ farmer(f);\ [d];\ donkey(d);\ owns(f,\ d)} h\}$

  $D_1.d$ includes free occurrences of f like a paycheck sentence!
- Therefore, $\mathbf{EVERY(D_1.d, D_2.d)}$ asserts that every donkey **d** that the current farmer **f** owns is such that **f** beats **d**.

References

van den Berg, M. H.: 1996, *Some aspects of the internal structure of discourse. The dynamics of nominal anaphora*, PhD thesis, University of Amsterdam.

Brasoveanu, A.: 2007, *Structured nominal and modal reference*, PhD thesis, Rutgers, The State University of New Jersey.

Geach, P.: 1962, *Reference and generality: an examination of some medieval and modern theories*, Ithaca, NY: Cornell University Press.

Groenendijk, J. and Stokhof, M.: 1991, Dynamic predicate logic, *Linguistics and philosophy* **14**(1), 39–100.

Hardt, D.: 1993, *Verb Phrase Ellipsis: form, meaning, and processing*, PhD thesis, University of Pennsylvania.

Hardt, D.: 1999, Dynamic interpretation of verb phrase ellipsis, *Linguistics and Philosophy* **22**(2), 187–221.

Karttunen, L.: 1969, Discourse referents, *Proceedings of the 1969 conference on Computational linguistics*, Association for Computational Linguistics, pp. 1–38.

Nouwen, R.: 2003, *Plural pronominal anaphora in context*, PhD thesis, University of Utrecht.

Roberts, C.: 1987, *Modal subordination, anaphora, and distributivity*, PhD thesis.