Lexical Threading

Guy Emerson

An Unforeseen Conflict

Emily M. Bender: “After 16 years of teaching grammar engineering, I thought I’d found all of the ways in which one can be in the situation of seemingly being able to build an edge through interactive unification which isn’t in the chart…

An Unforeseen Conflict

Emily M. Bender: “After 16 years of teaching grammar engineering, I thought I’d found all of the ways in which one can be in the situation of seemingly being able to build an edge through interactive unification which isn’t in the chart…

“Alas, I’ve found evidence of a new one.”

Coordinating Monkeys

The monkeys ate bananas and slept

The monkeys ate bananas quickly and slept

An Unforeseen Conflict

- Pick at most 3 out of:
  - Lexical threading of SLASH
  - Argument/adjunct distinction
  - Identifying structures containing lists
  - Append-lists

An Unforeseen Conflict

- Pick at most 3 out of:
  - Lexical threading of SLASH
  - Argument/adjunct distinction
  - Identifying structures containing lists
  - Append-lists
- *(Not originally on slides: another issue is whether we can rely on deleted-daughters)*

An Unforeseen Conflict

- Lexical threading performs list appends early (the SLASH of a phrase is the SLASH of the head)
- Adjuncts can perform further appends
- Phrases with the same valence may have used different numbers of appends
- Append-lists keep the number of appends

Which Analysis to Drop?

- Lexical threading: instead, perform appends at the point of composition
- Argument/adjunct: instead, treat all adjuncts as arguments
- Identifying append-lists: instead, keep “clean” copy of lists separate from append machinery, and only ever identify clean lists
- Append-lists: instead, stick to diff-lists, and use other means to check list length

Drop Lexical Threading?

- Rationale:
  - Lexical threading identifies a phrase’s SLASH with the head’s SLASH
  - Unlike COMPS, can’t be seen as an input/output process: the head’s SLASH is underspecified until it has picked up all its arguments
  - Underspecification is harder to reason about, and awkward for defining a composition algebra…

Drop Lexical Threading?

- Rationale:
  - Lexical threading identifies a phrase’s SLASH with the head’s SLASH
  - Unlike COMPS, can’t be seen as an input/output process: the head’s SLASH is underspecified until it has picked up all its arguments
  - Underspecification is harder to reason about, and awkward for defining a composition algebra…
- Proposal:
  - Lexically control how SLASH will be used
  - Append at the point of composition

Drop Argument/Adjunct?

- Bouma et al. DEPS

Drop Identifying Append-lists?

- Use two features: LIST and DIFF-LIST
- LIST contains a clean list, which can be identified
- DIFF-LIST contains a diff-list, which can be appended to

Drop Append-lists?

- Diff-lists don’t have this problem
- To access elements, use the LIST
- To check length, use subtypes of diff-list

Previous discussions

- https://delphinqa.ling.washington.edu/t/edge-can-be-built-interactively-but-isnt-in-the-chart/427
- https://delphinqa.ling.washington.edu/t/an-unforeseen-conflict-lexical-threading-adjuncts-coordination-and-append-lists/438