

Discourse Parsing

Natalie Parde

UIC CS 421

Given a specified discourse model (e.g., RST), how do we automatically assign discourse relations to text?

- **Discourse structure parsing:** Given a sequence of text, automatically determine the coherence relations between spans within it
- Discourse structure parsing can be performed similarly to constituency parsing
 - Break text into meaningful subunits
 - Organize those subunits into a set of directed (and, depending on model type, hierarchical) relations



What does this look like for RST parsing?

- **Step #1: EDU Segmentation**

- Extract the start and end of each elementary discourse unit

Natalie said there were no office hours on Thursday because it was Thanksgiving.



[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}



EDU Segmentation



- EDUs roughly correspond to clauses
- Early EDU segmentation approaches:
 - Run a syntactic parser
 - Post-process the output
- More modern EDU segmentation approaches:
 - Usually, apply supervised neural sequence models



What does this look like for RST parsing?

- **Step #1: EDU Segmentation**
 - Extract the start and end of each elementary discourse unit
- **Step #2: Parsing Algorithm**
 - Build representations for each EDU, and apply some method to connect them using RST relations

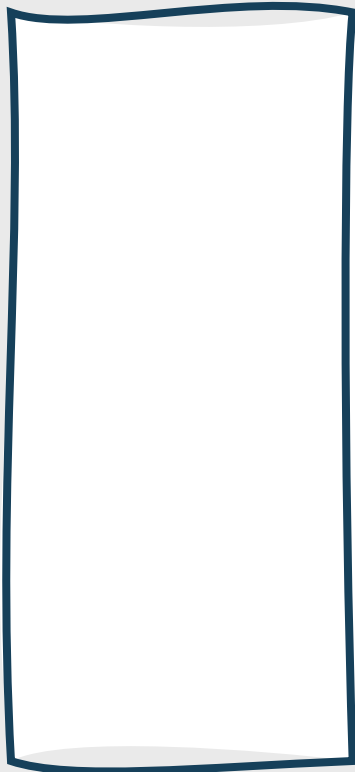
RST Parsing

- Generally based on syntactic parsing algorithms
- Common syntactic parsing approach: **Shift-reduce parser**
 - **Shift:** Push an EDU from the queue onto the stack, creating a single-node subtree
 - **Reduce:** Merge the top two subtrees (either single-node or more complex) on the stack, assigning a coherence relation label and a nuclearity direction
 - **Pop:** Remove the final tree from the stack

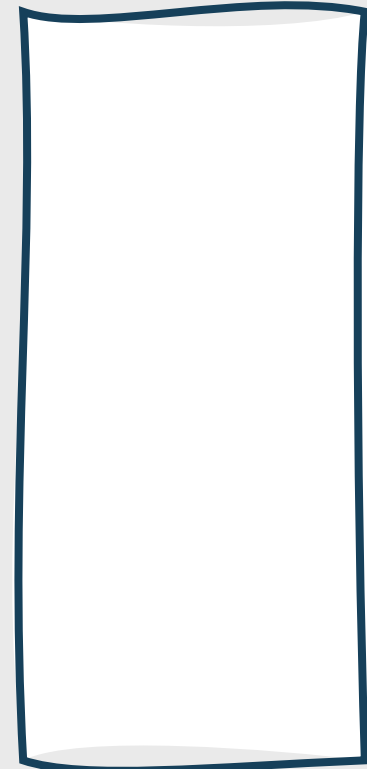
Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}

Queue



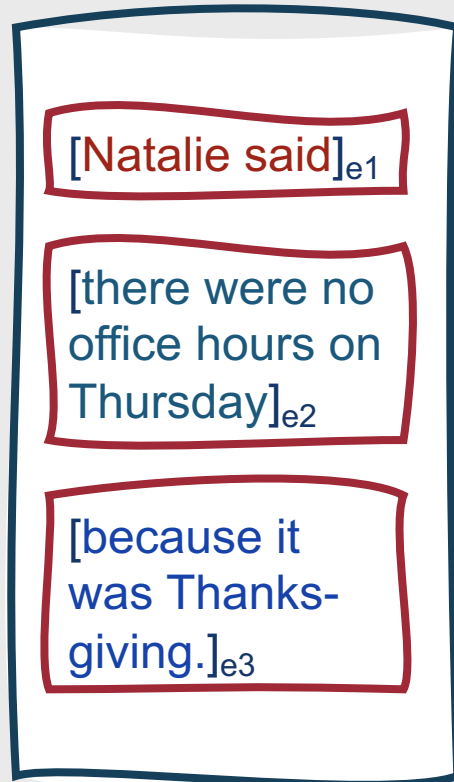
Stack



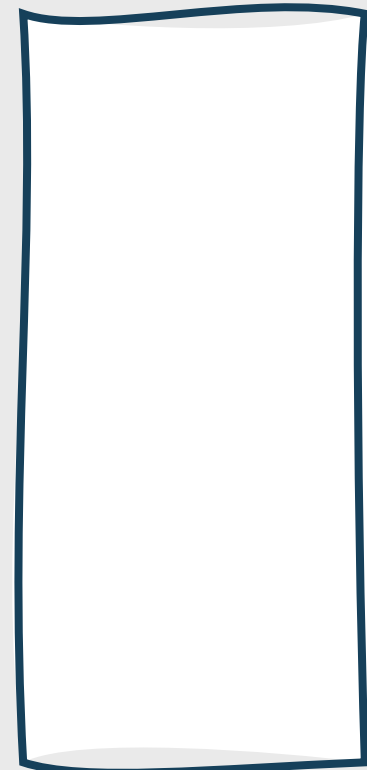
Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}

Queue

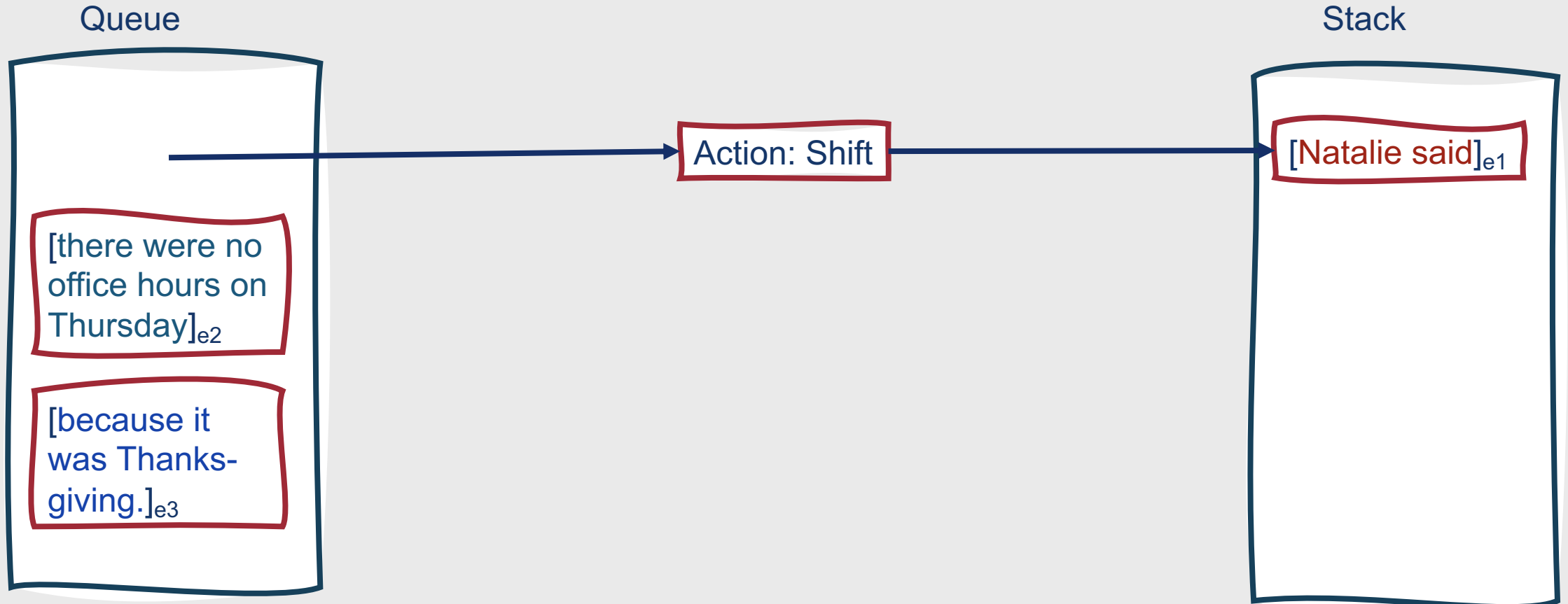


Stack



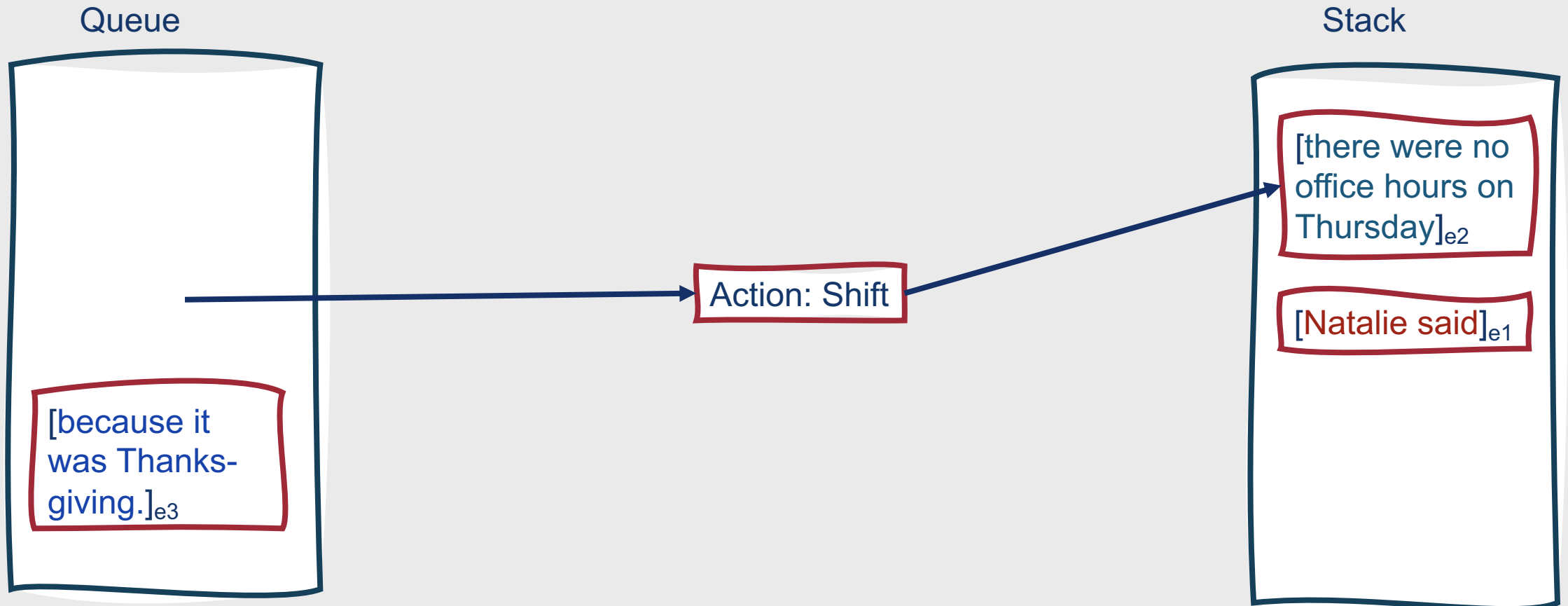
Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}



Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}



Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}

Queue



Stack

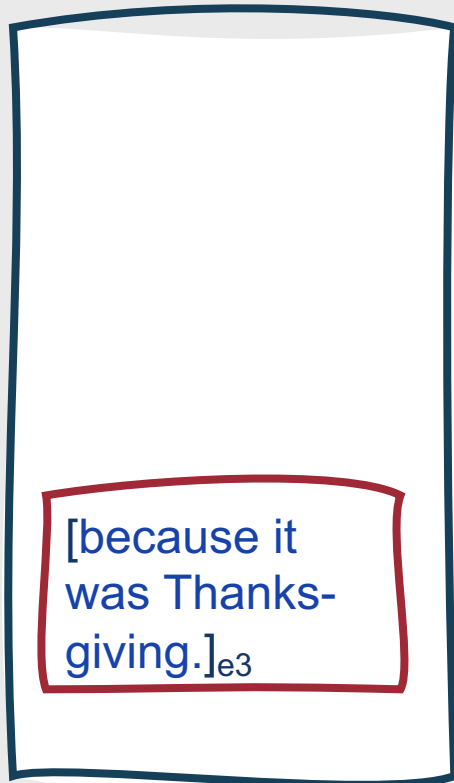


Action: Reduce(Attribution, (Satellite, Nucleus))

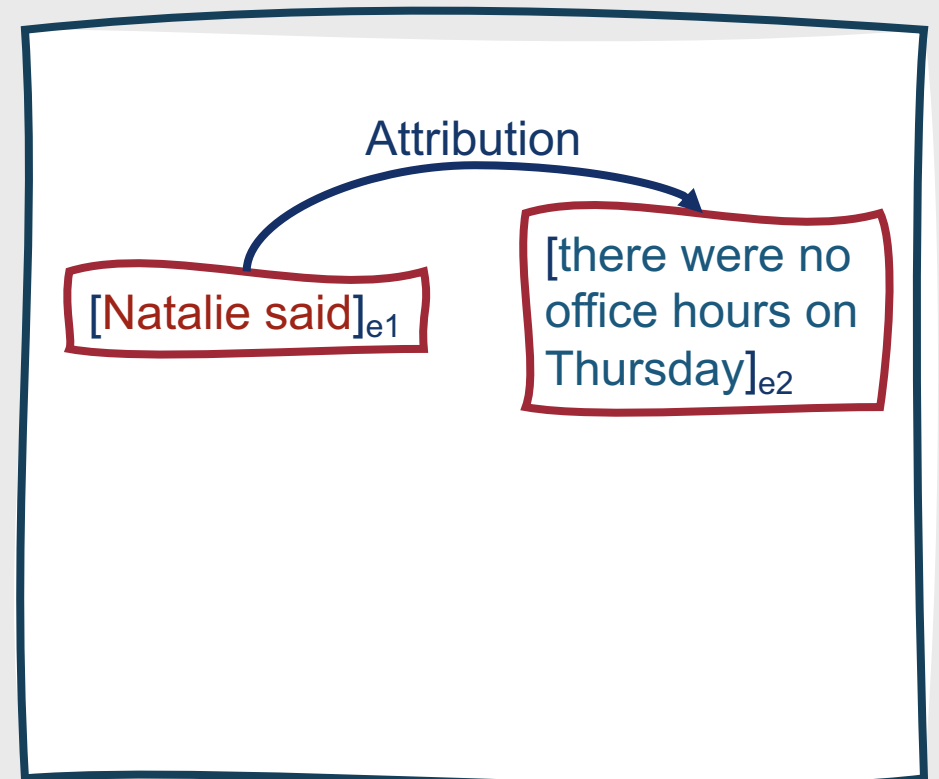
Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}

Queue

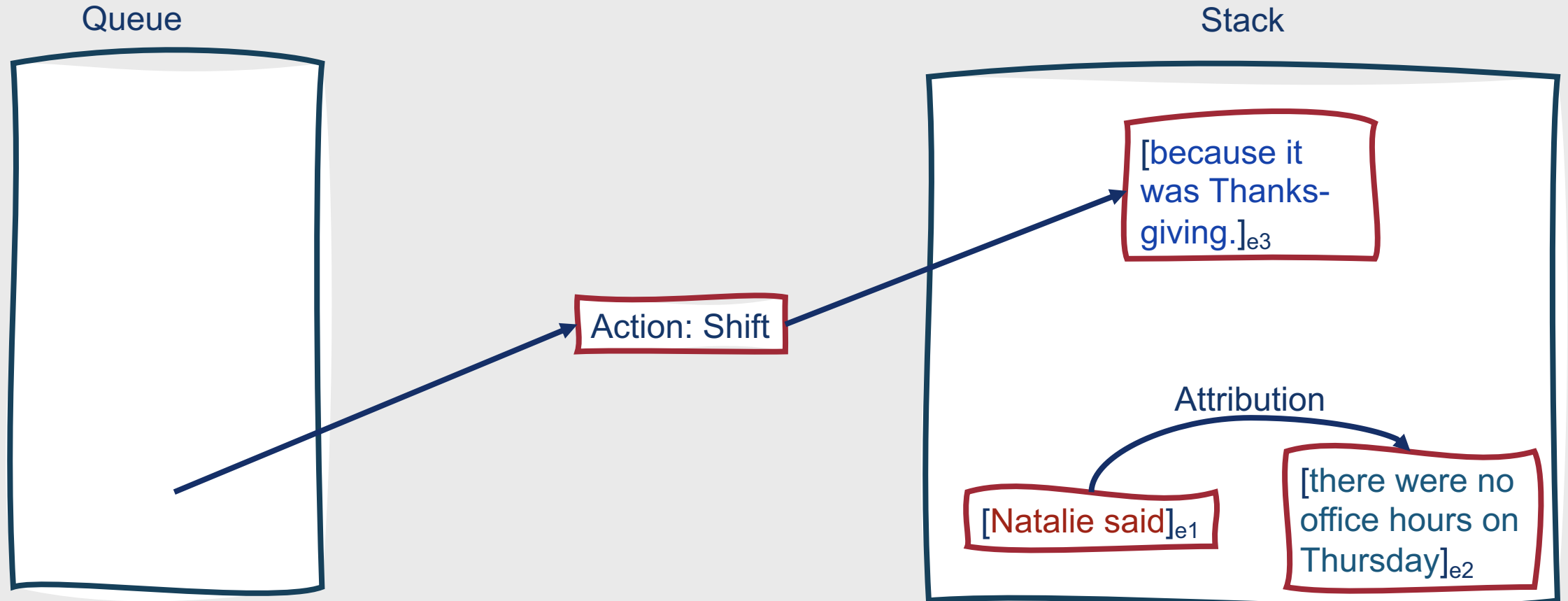


Stack



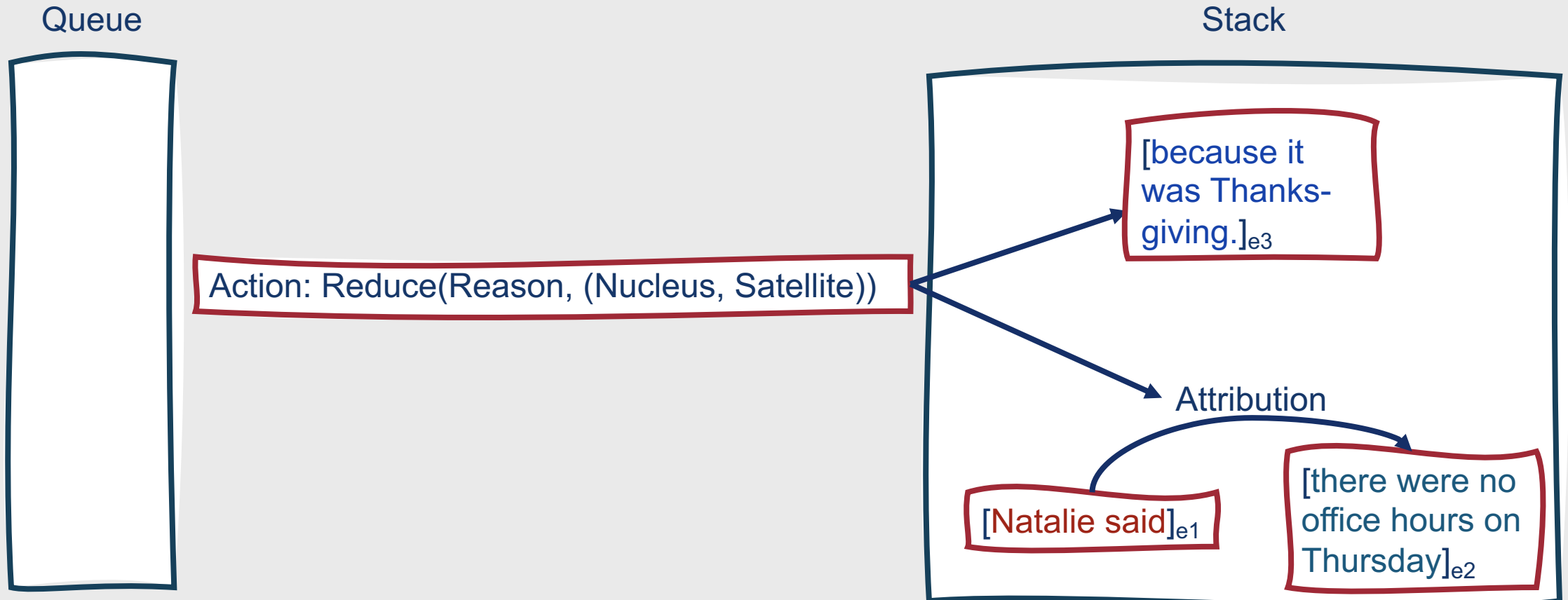
Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}



Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}

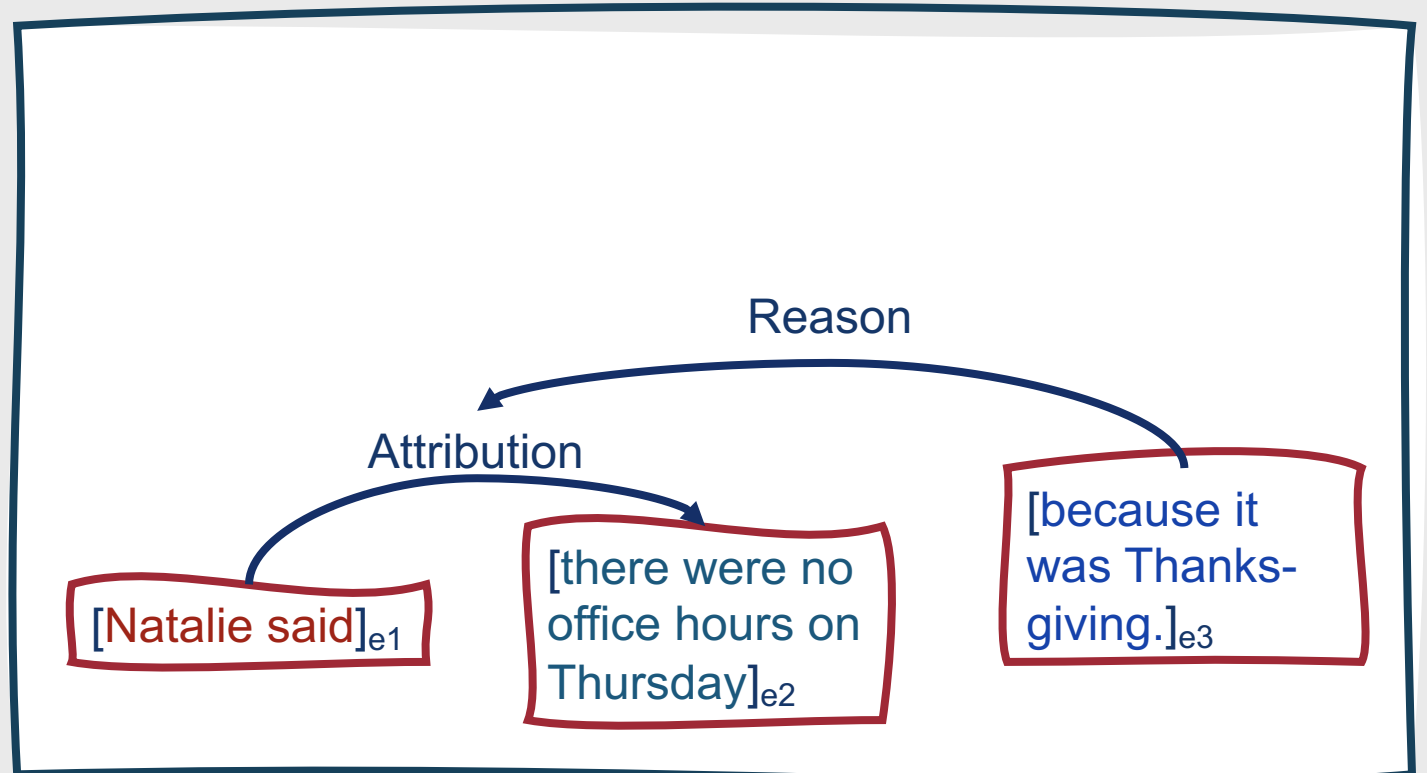
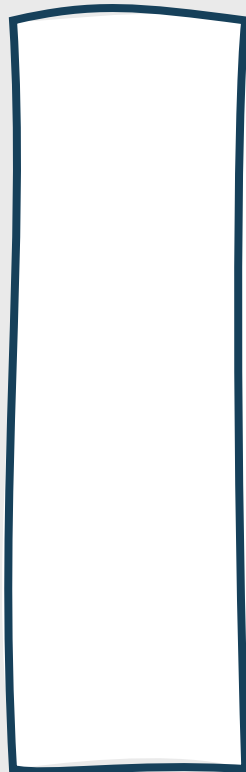


Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}

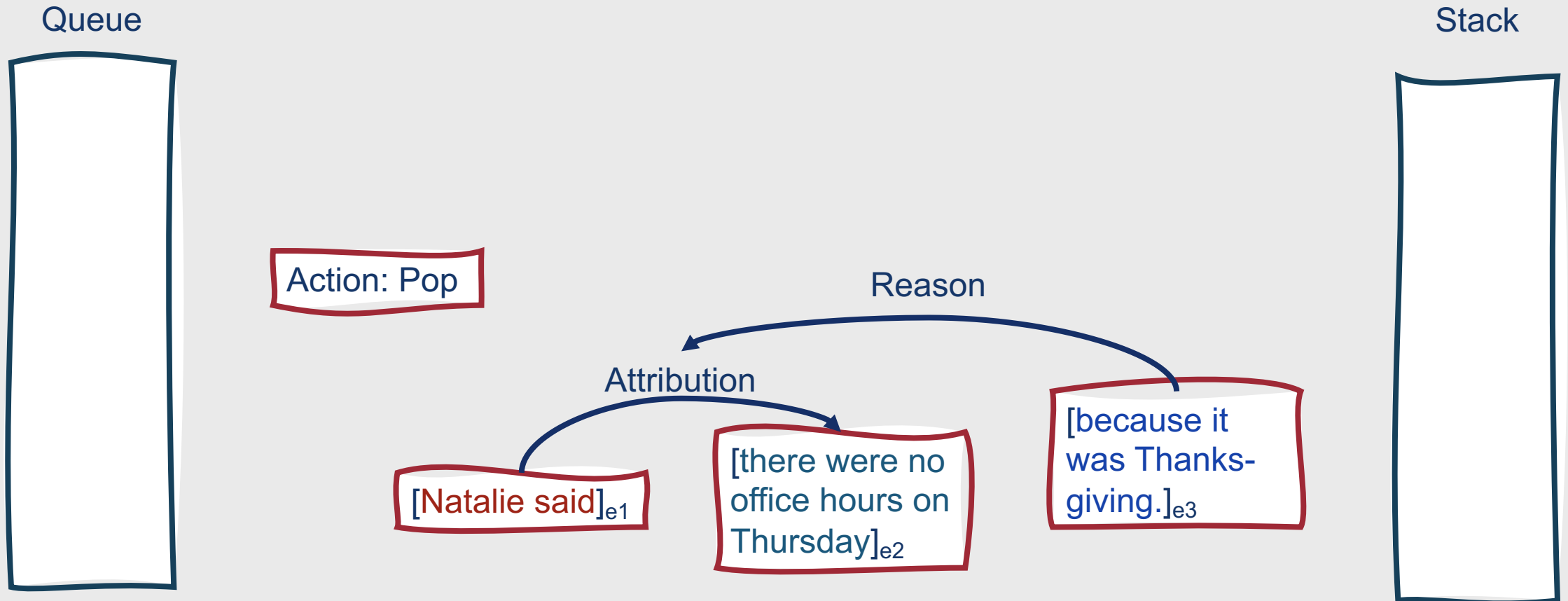
Queue

Stack

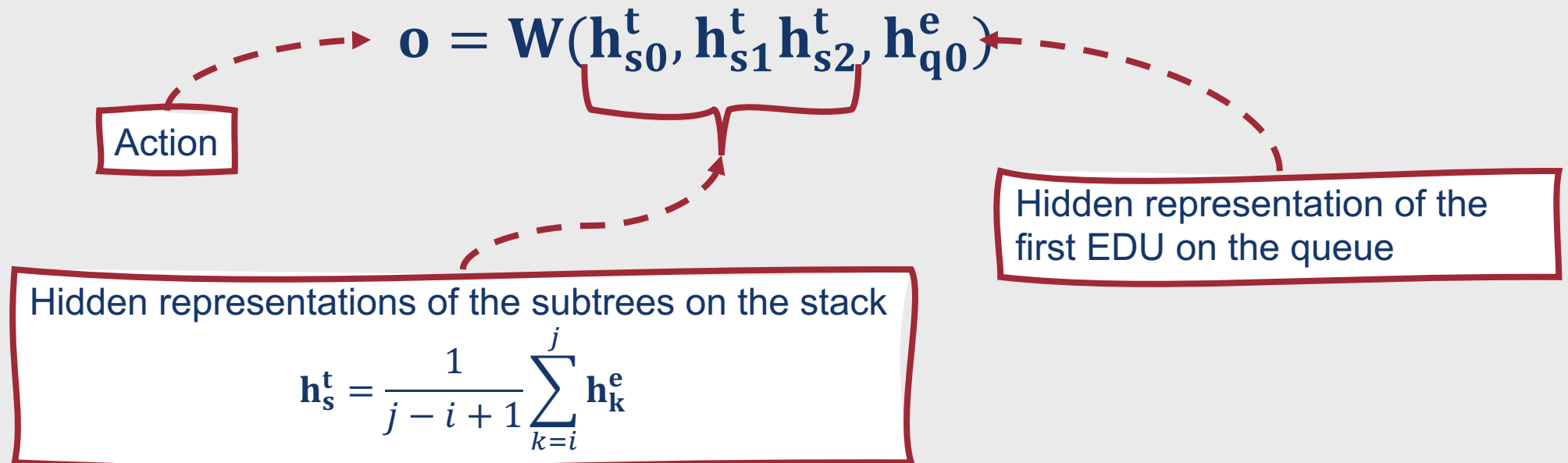


Example: Shift-Reduce Parser

[Natalie said]_{e1} [there were no office hours on Thursday]_{e2} [because it was Thanksgiving.]_{e3}



Modern RST parsers generally select actions using neural networks.



How does PDTB discourse parsing differ from this?

- **Shallow discourse parsing:** Identifying relationships between text spans only, rather than full hierarchical discourse trees