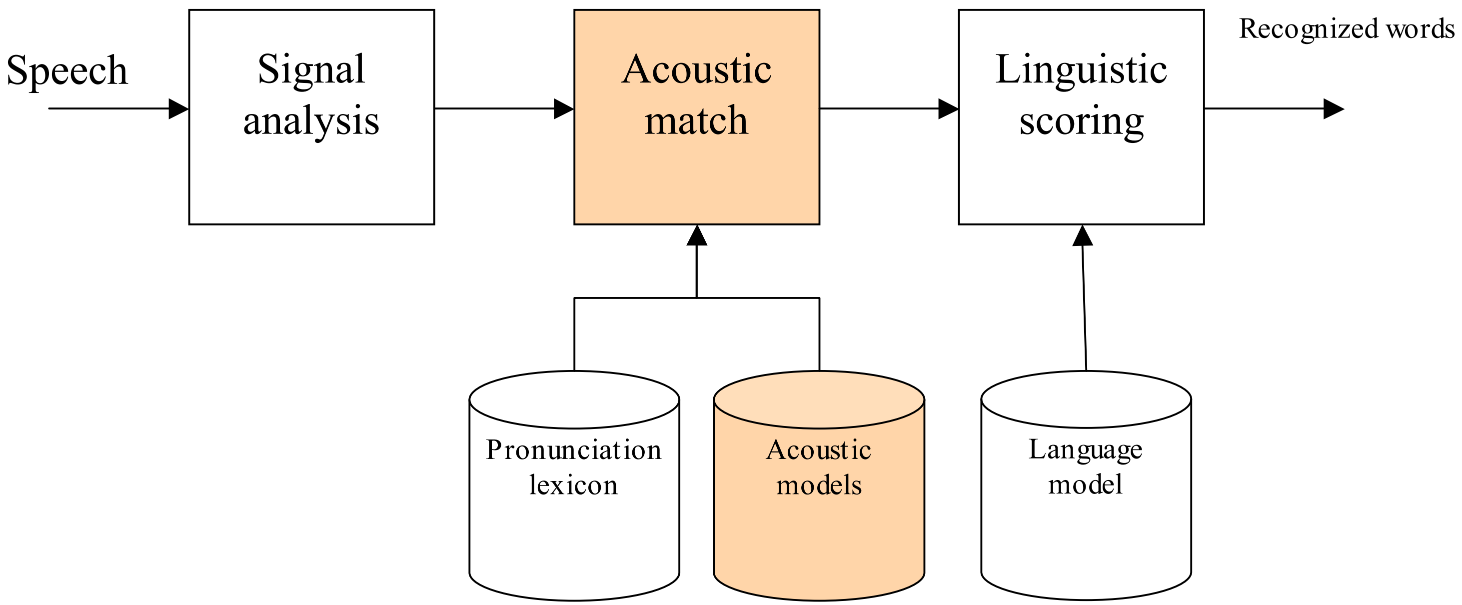
Acoustic match - templates: Outline

- Template based pattern matching
- Dynamic time warping
- Dynamic programming

ASR step-by-step: Acoustic match (1)



Template based pattern matching

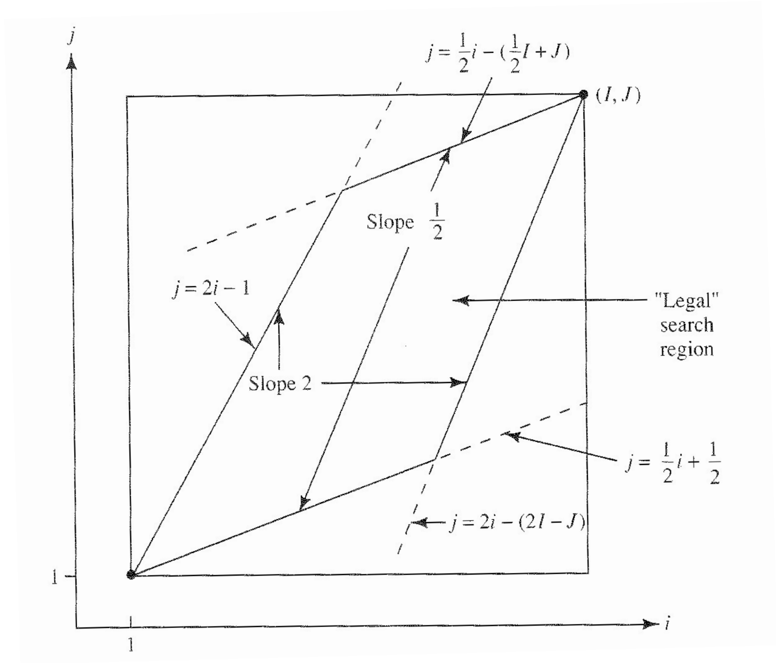
- Speech *recognition* implies that a pattern has already been learned
  - Training
- In template matching techniques, the learned pattern is represented as a temporal pattern, e.g. a (typical) sequence of feature vectors
- Recognition basically consists of evaluating the match between the test pattern and the stored patterns and selecting the closest matching stored pattern as the recognized pattern
- The speech patterns will exhibit relativly large temporal variations
  - Non-linear dependency on speaking rate
- How to account for ”normal” temporal variations?
- Dynamic Time Warping (Sakoe and Chiba, 1978)

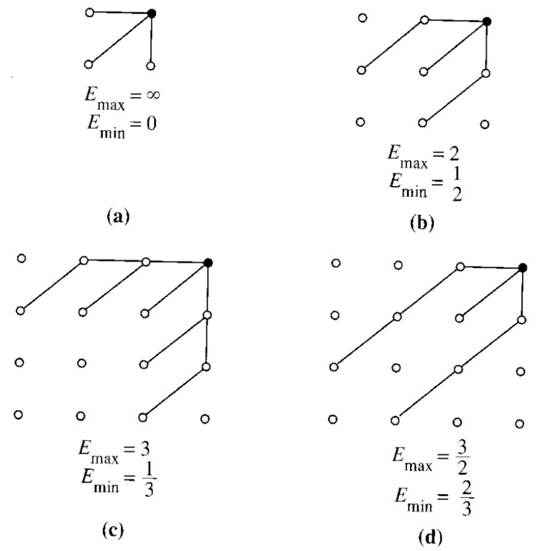
NTNU

Dynamic Time Warping

- Method for aligning two temporal pattern series
- Based on Dynamic programming (Bellmann, 1957)
- Requires a metric for local distance, i.e. a measure of the dissimilarity between two feature vectors, $d(\mathbf{x},\mathbf{y})$
  - Should be meaningful
  - $d(\mathbf{x},\mathbf{x})=0$
  - $d(\mathbf{x},\mathbf{y})>0$ iff $\mathbf{x}\neq\mathbf{y}$
  - $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$ (symmetry - desirable, not necessary)

Global and local constraints





Global constraints

Local constraints

- Restrict freedom of search to better correspond with natural temporal variations of speech whilst containing the left-right ordering of acoustic events

Dynamic programming

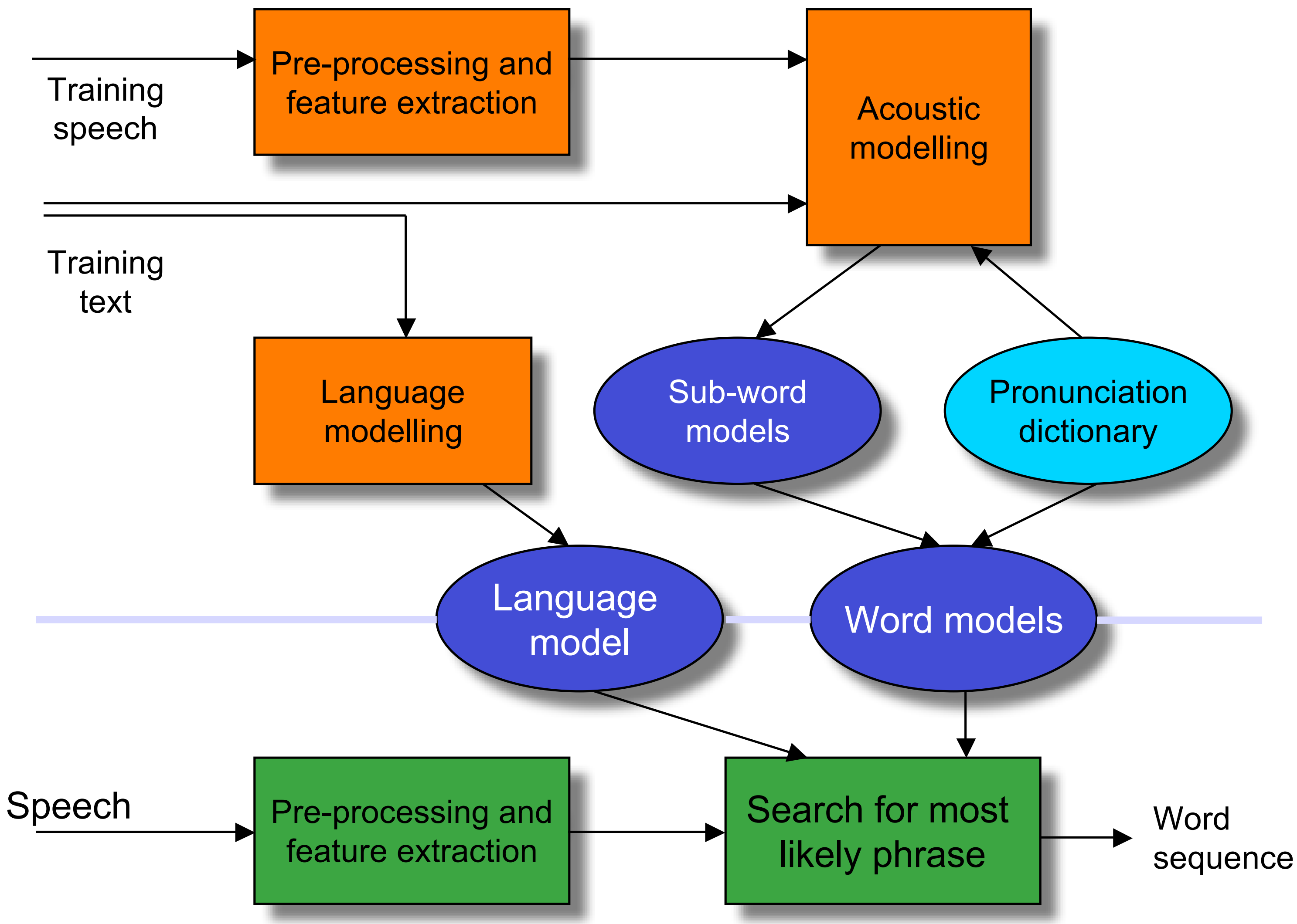
- Efficient method for search and matching
- Used in many ASR applications
- DTW: Given two sequences, $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_j\}$, i=1,...,N; j=1,...,M.
  - Find the warping, w(j), such that the total distance

$$\mathrm{D}(\mathbf{X},\mathbf{Y}) = \sum_i d(\mathbf{x}_i, \mathbf{y}_{w(j)})$$

is minimized

- Based on Bellmann's principle: If the optimal path between (1,1) and (N,M) passes through (n,m), then the optimal path between (1,1) and (n,m) is a part of the overall optimal path.
  - Can evaluate iteratively instead of searching through all possible paths
  - Optimal path to (n,m) can be found by evaluating accumulated distance at all immediate predecessors of (n,m) (plus a transition cost). Accumulated cost at (n,m) is found by adding local distortion.

ASR overview

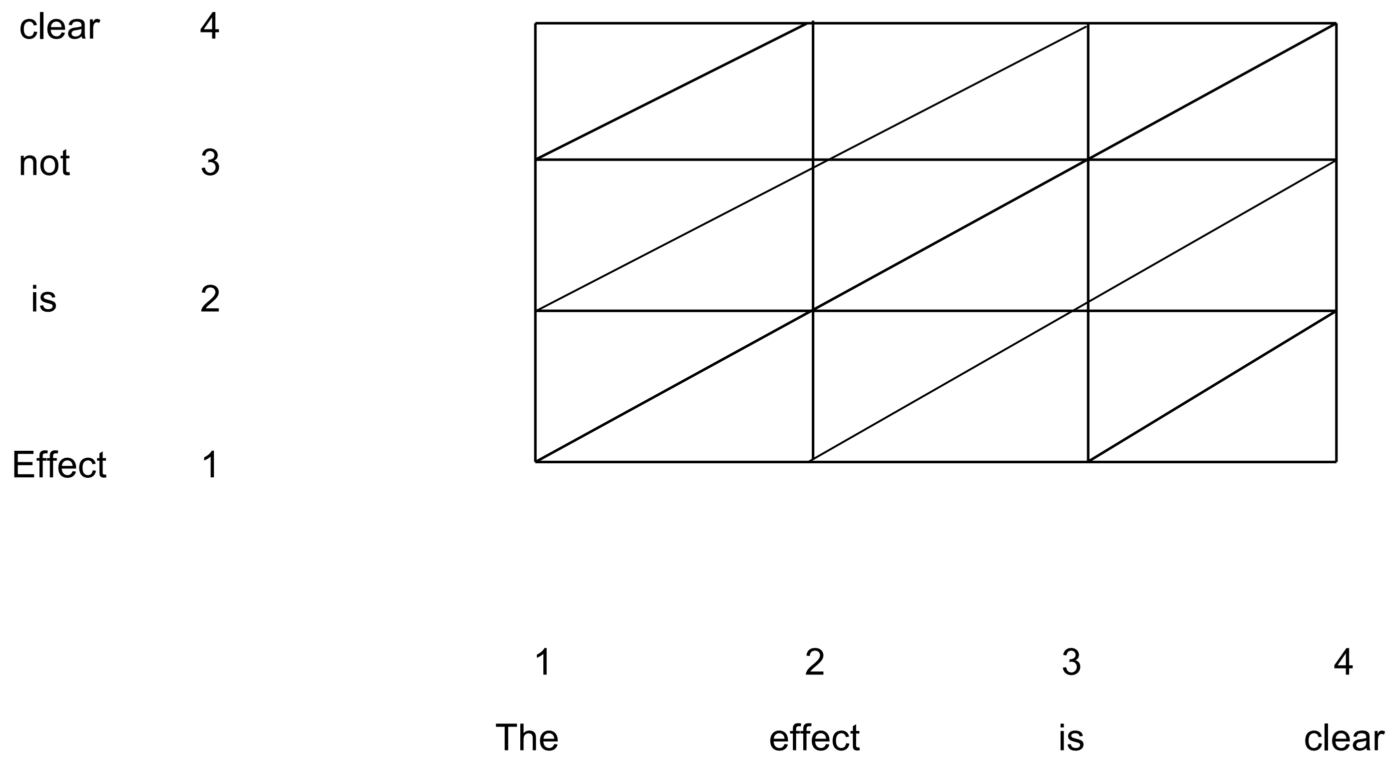


Dynamic programming example

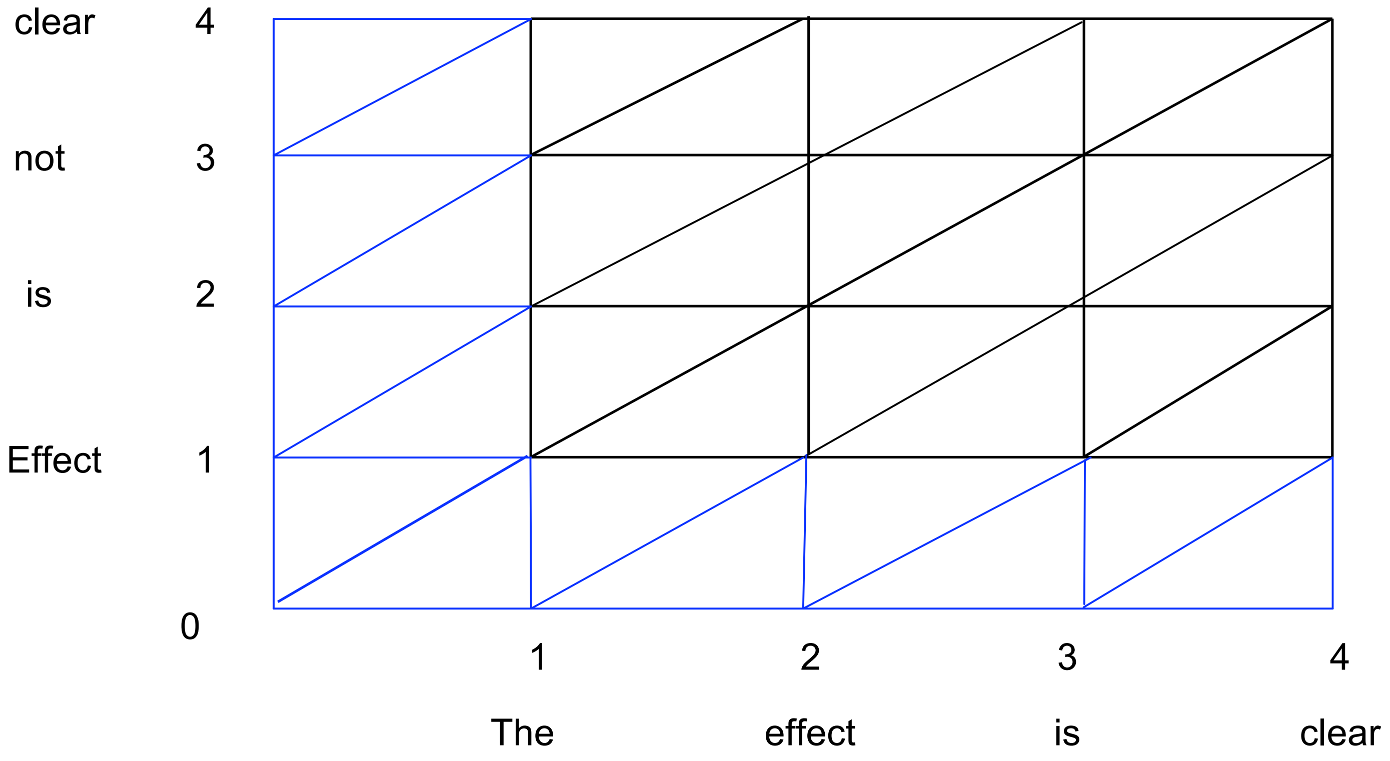
- Match two word sequences (e.g. spoken and recognized)
- Spoken: ”The effect is clear”
- Recognized: ”Effect is not clear”
- Penalty factors in dynamic programming
  - Deletion: $P_D=3$
  - Insertion: $P_I=3$
  - Substitution: $P_S=4$

NTNU

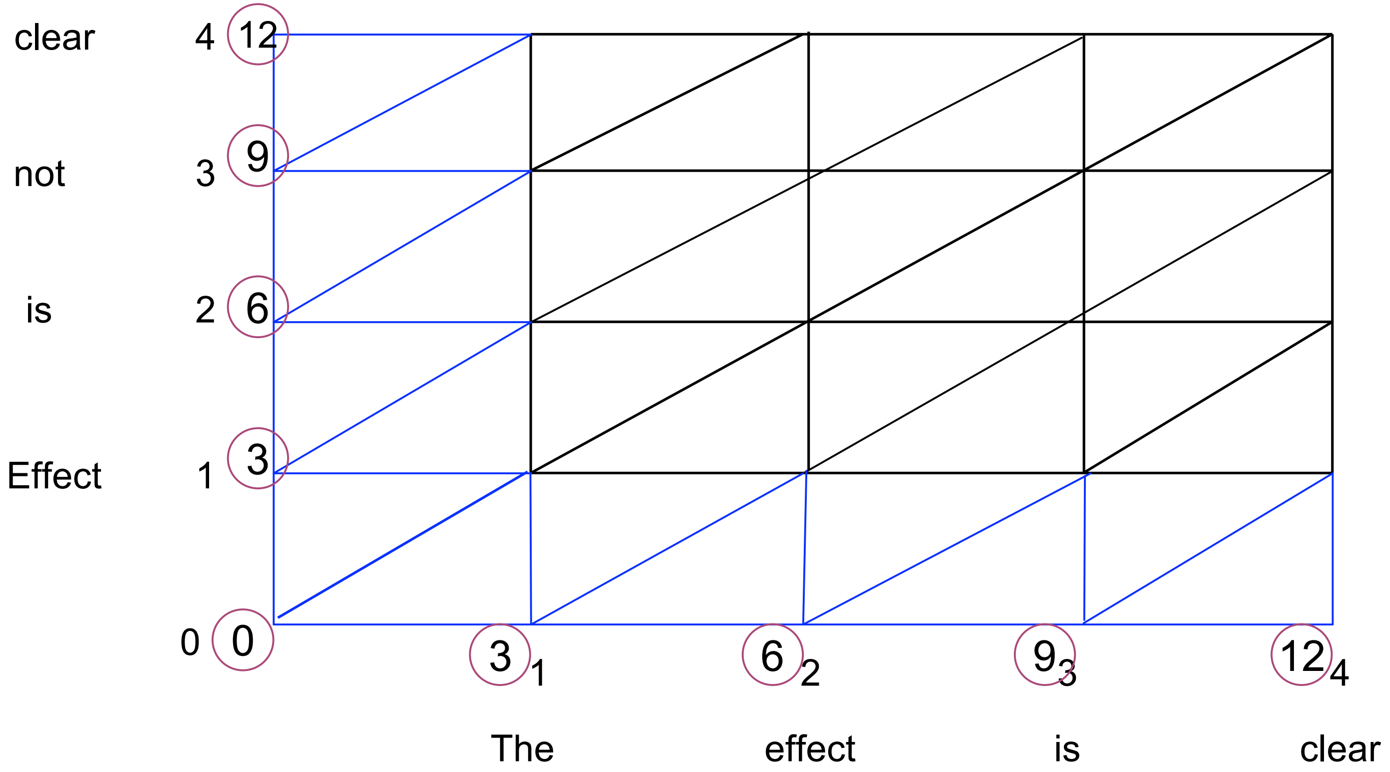
Dynamic programming example



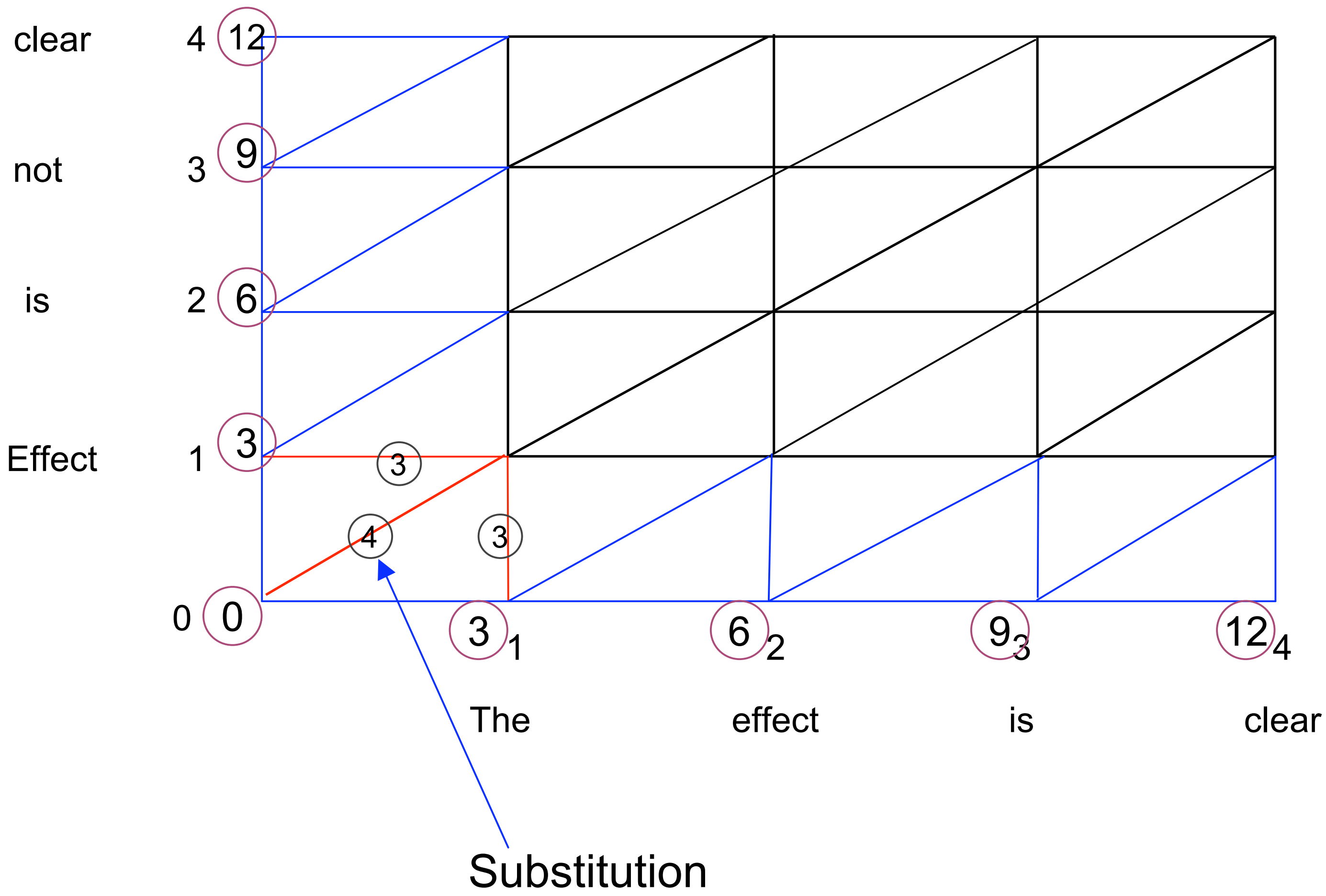
Dynamic programming example



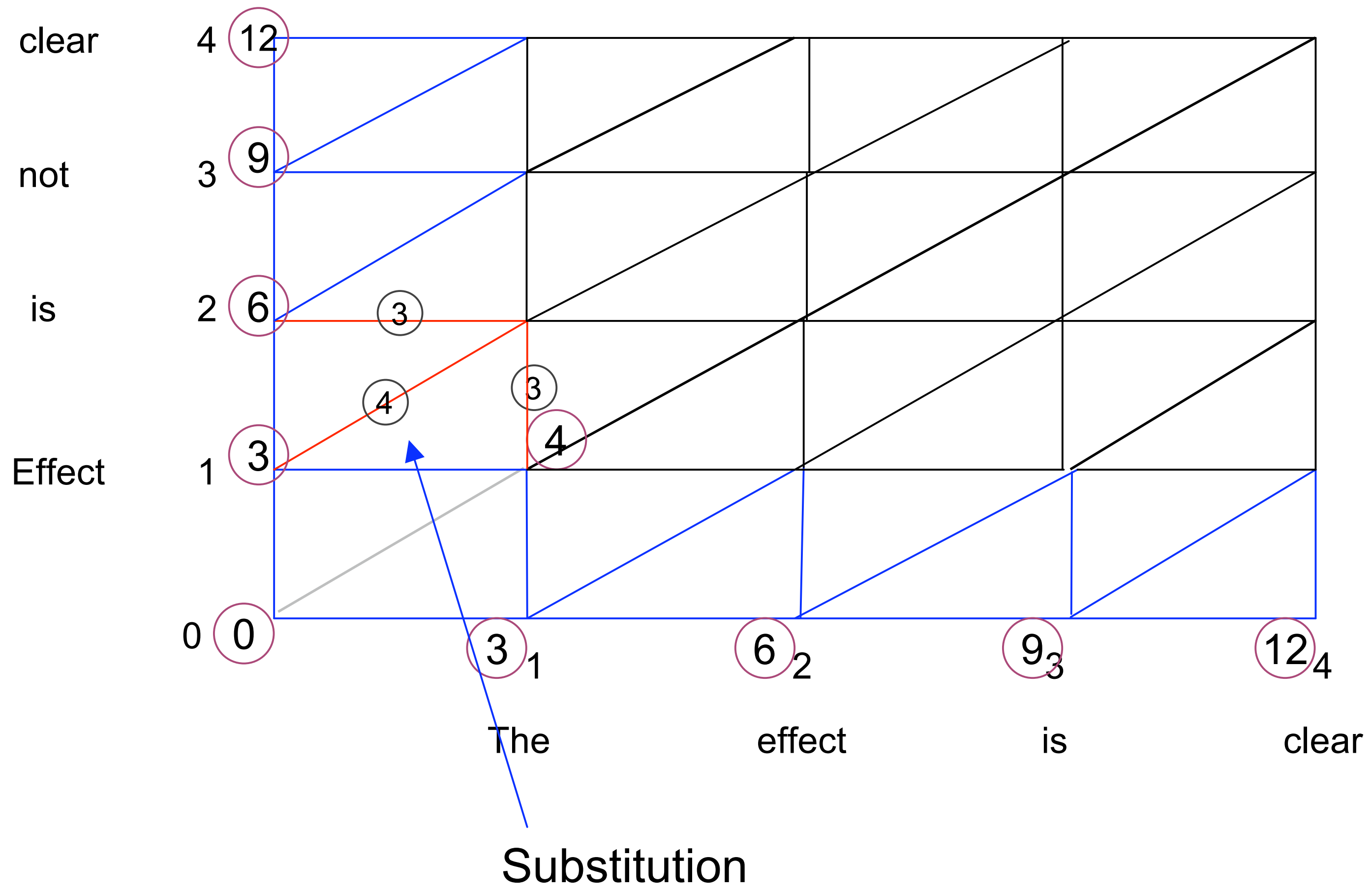
Dynamic programming example



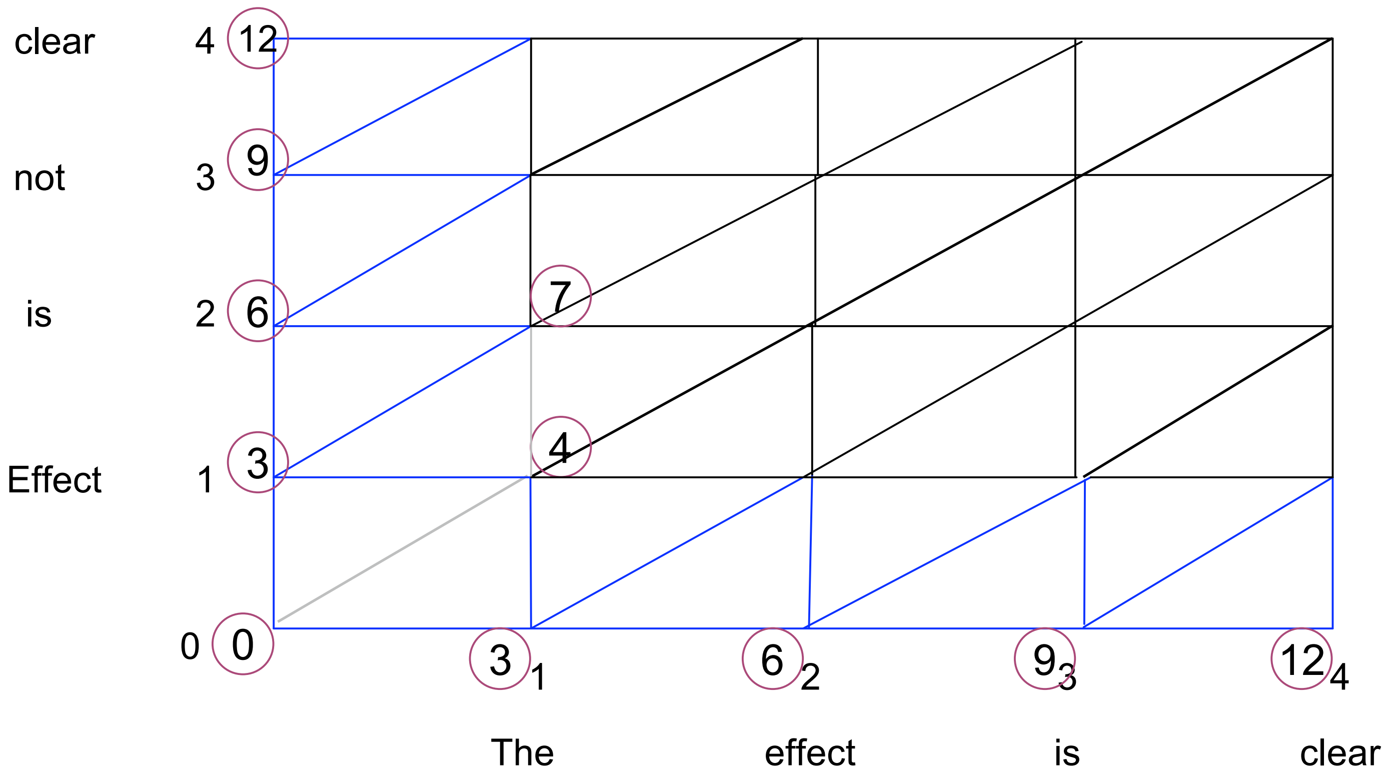
Dynamic programming example



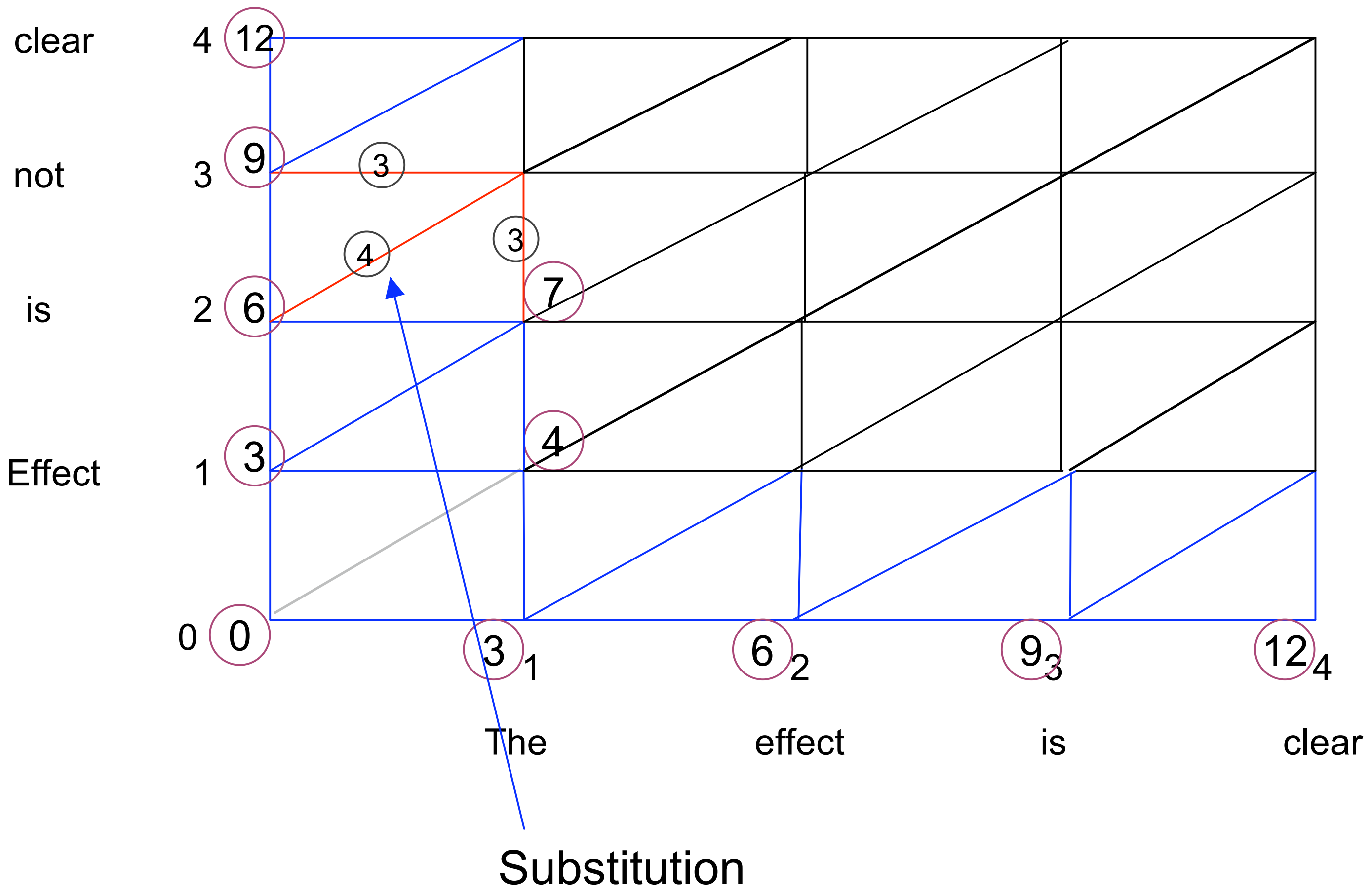
Dynamic programming example



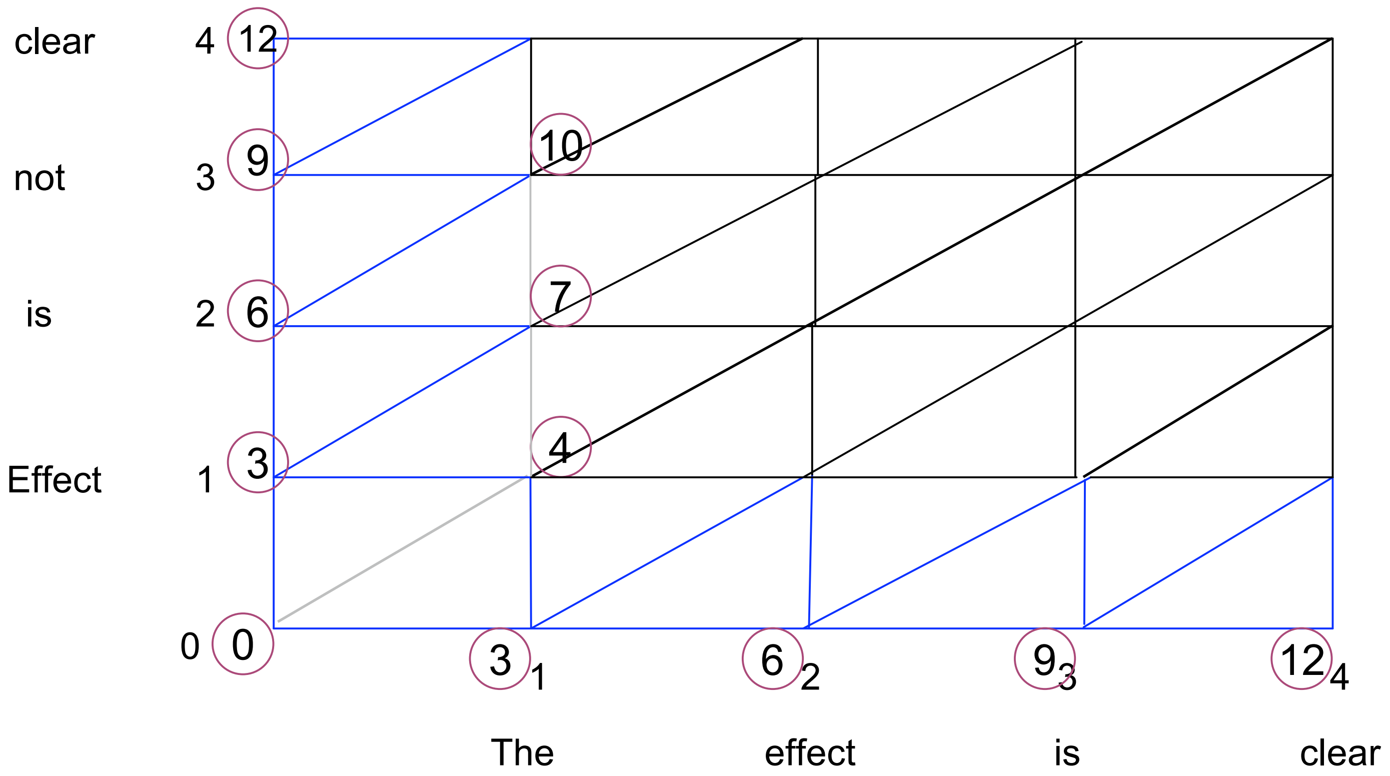
Dynamic programming example



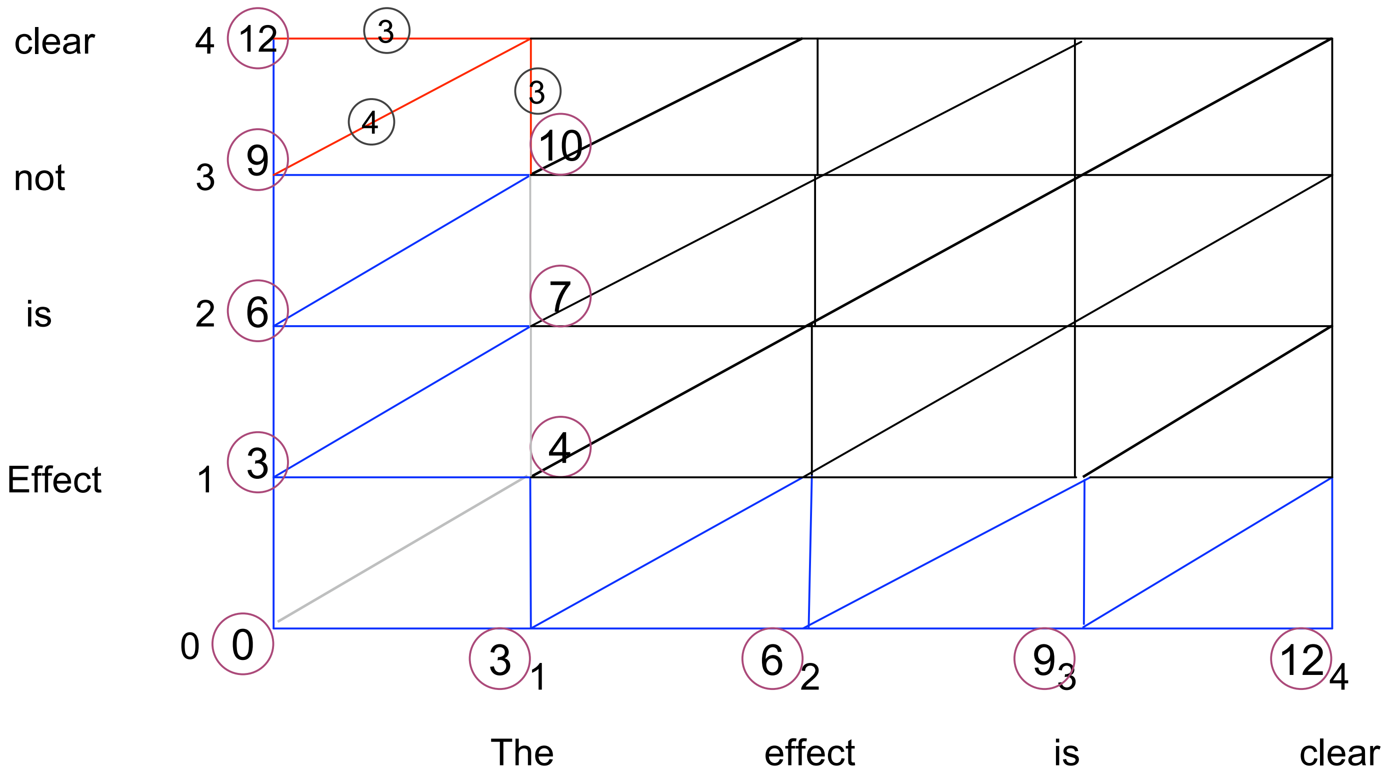
Dynamic programming example



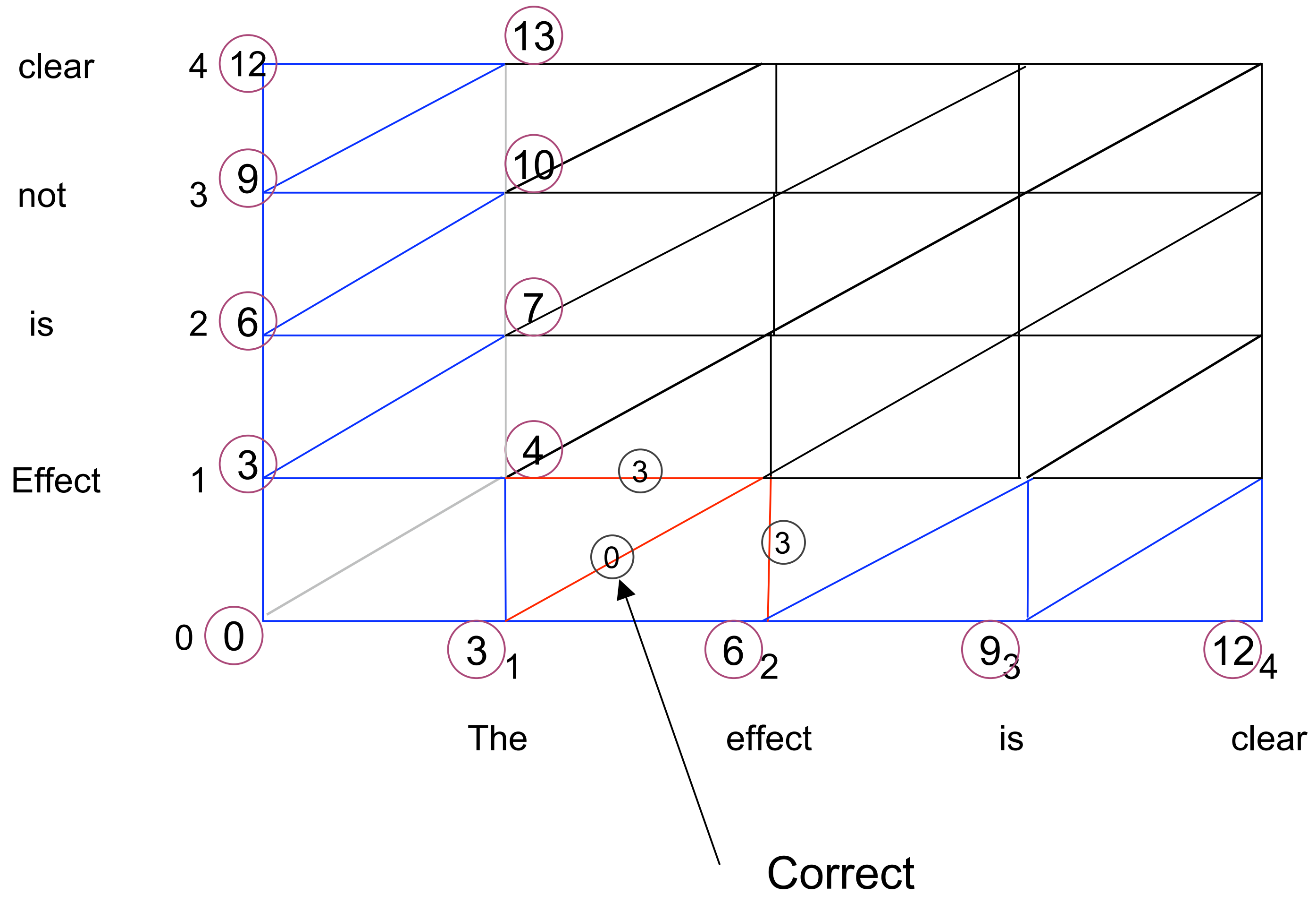
Dynamic programming example



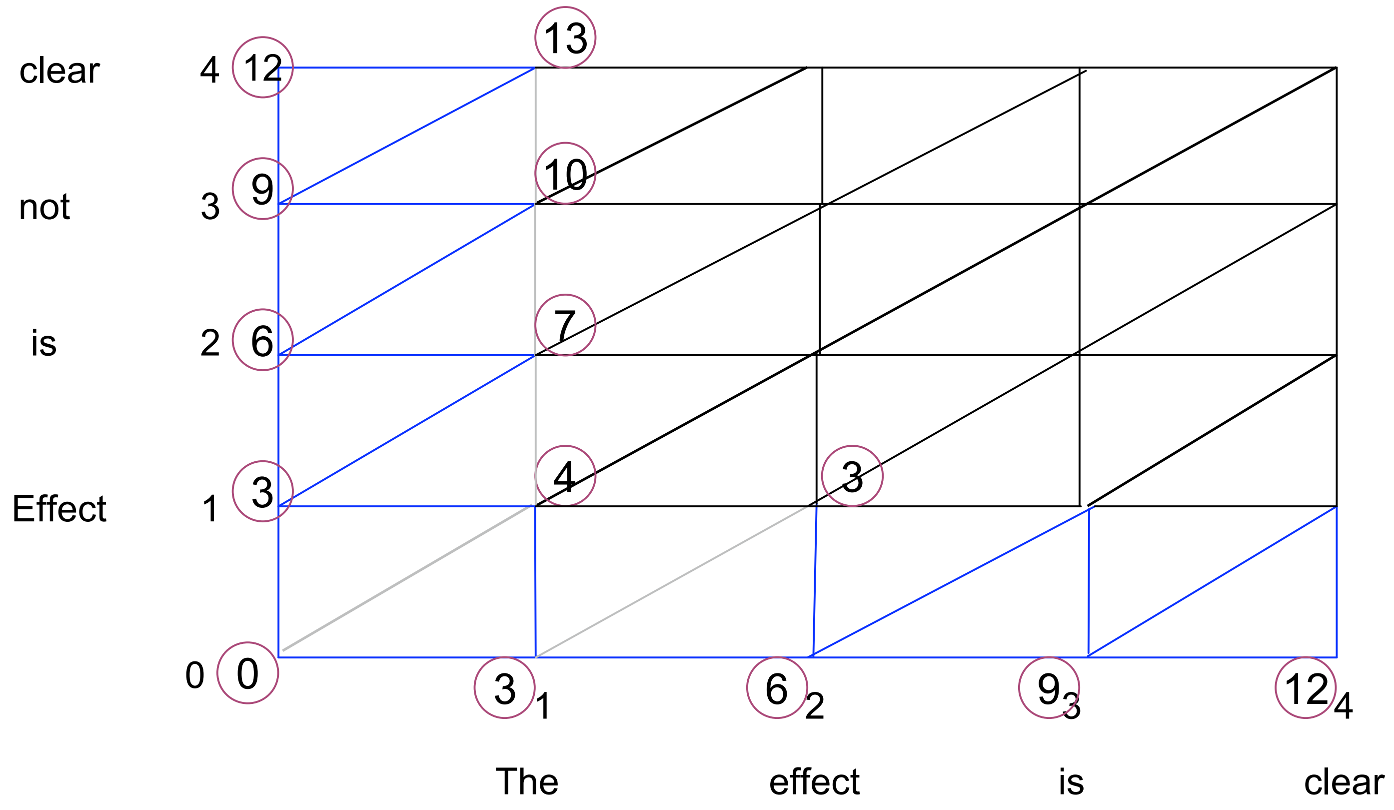
Dynamic programming example



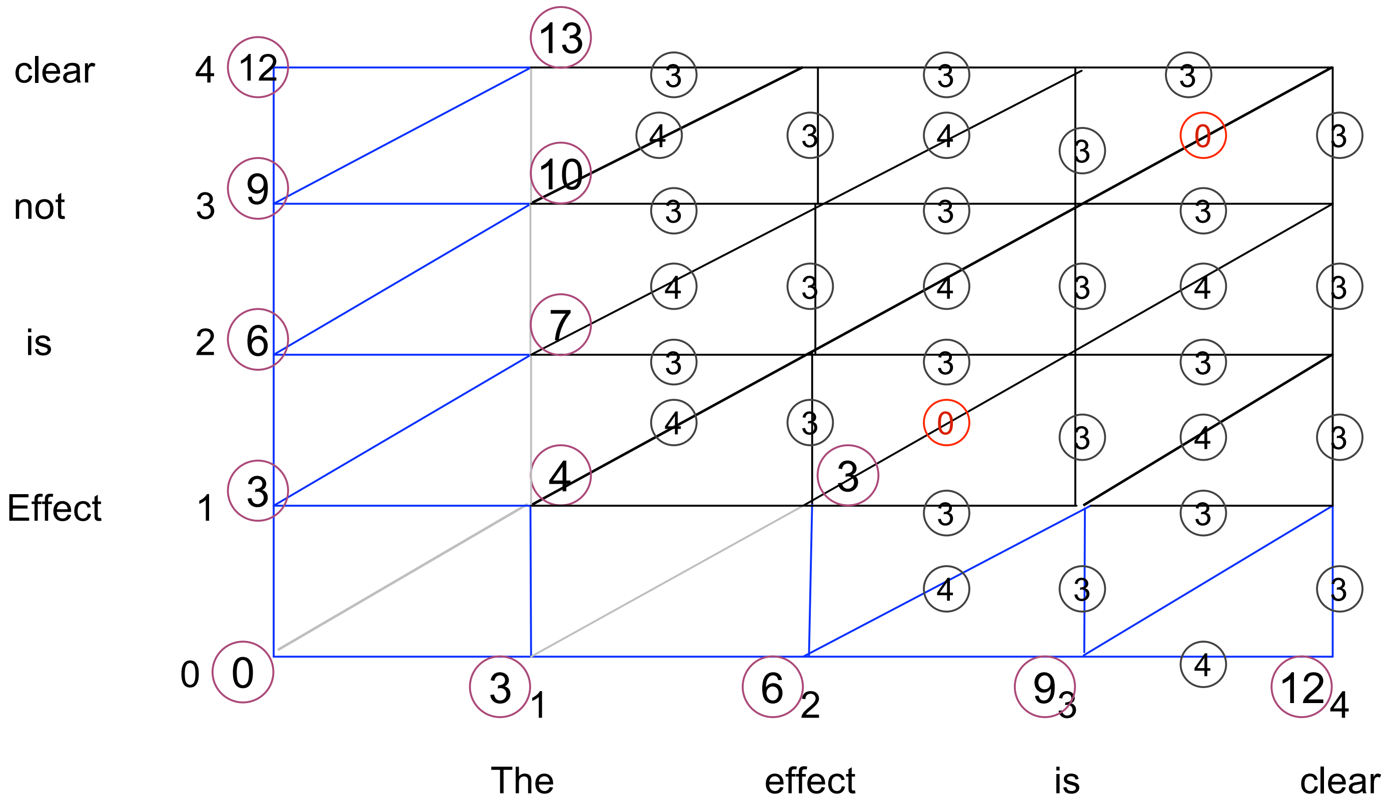
Dynamic programming example



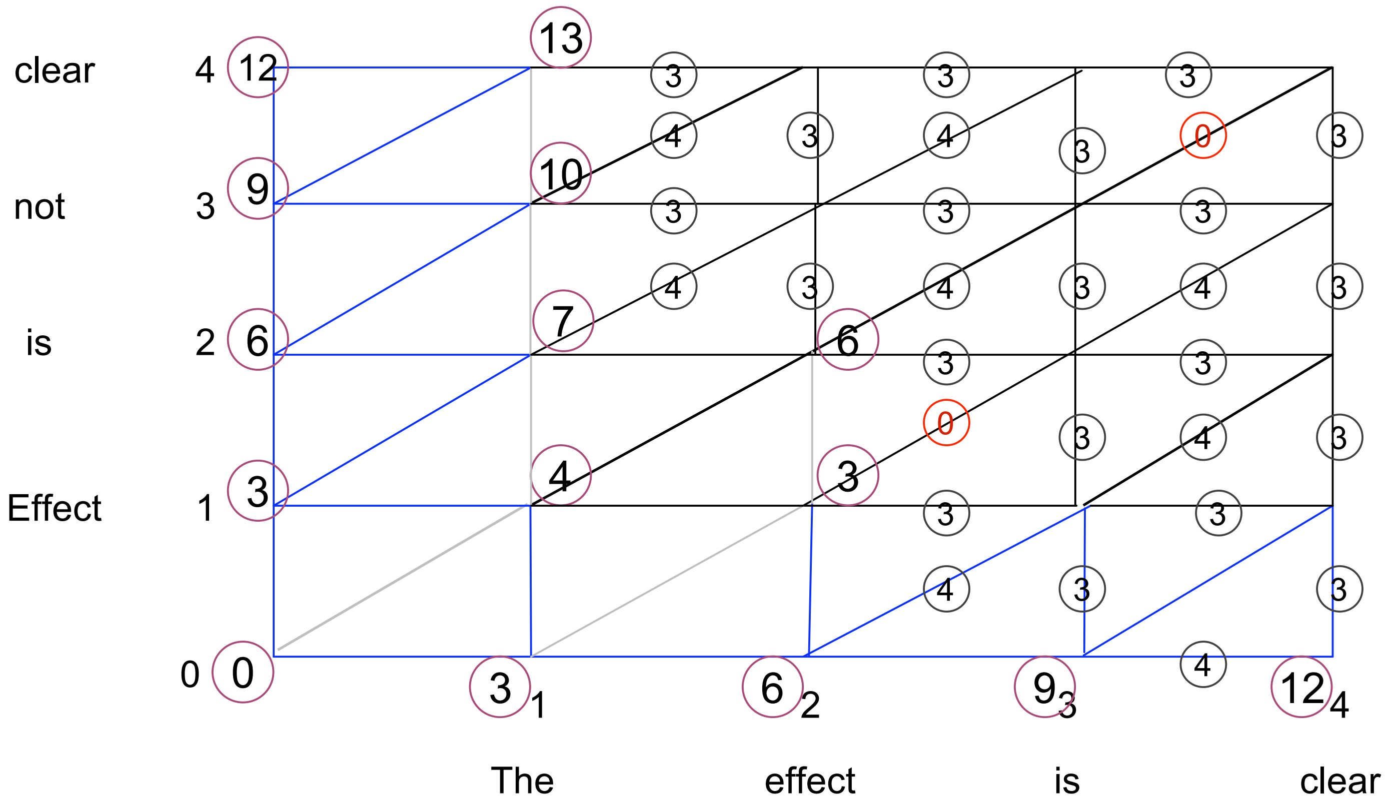
Dynamic programming example



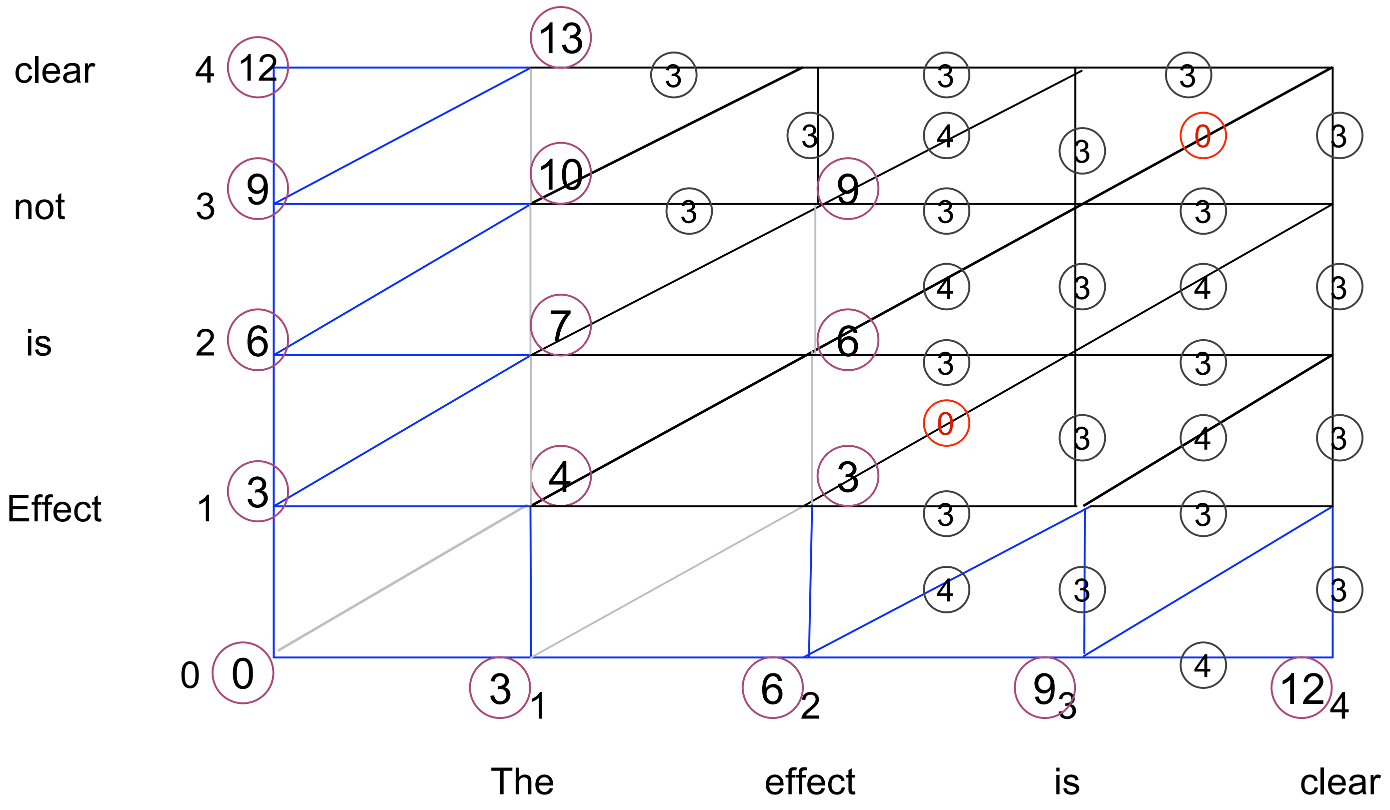
Dynamic programming example



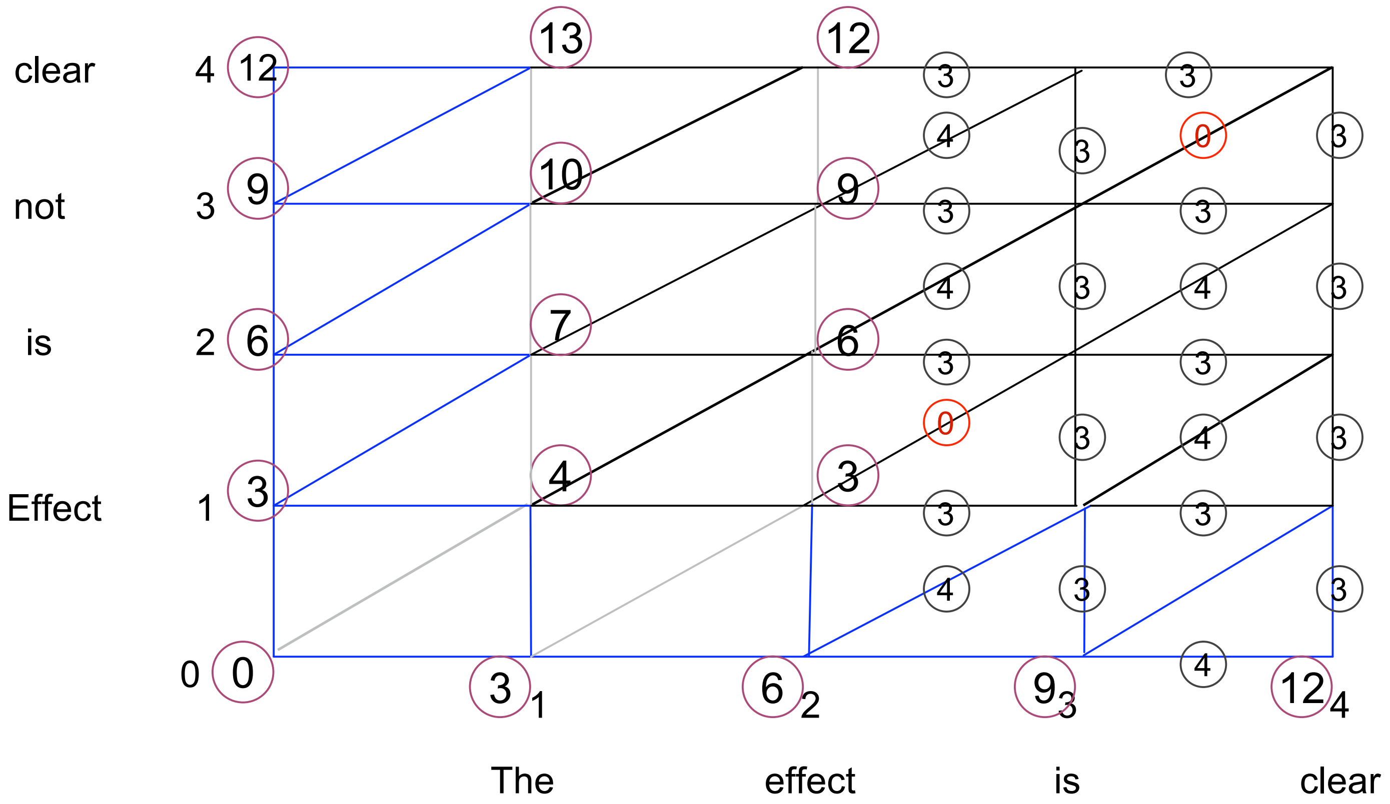
Dynamic programming example



Dynamic programming example

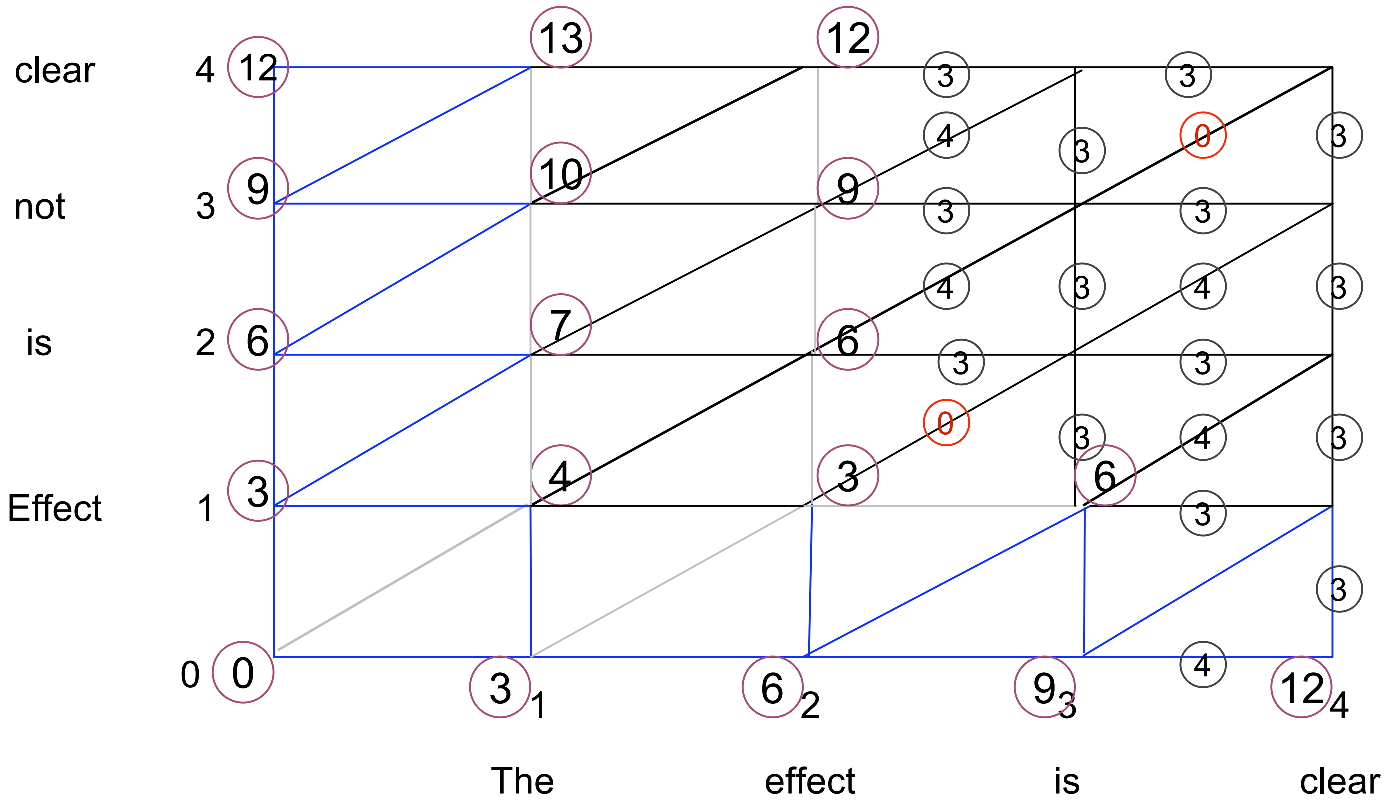


Dynamic programming example

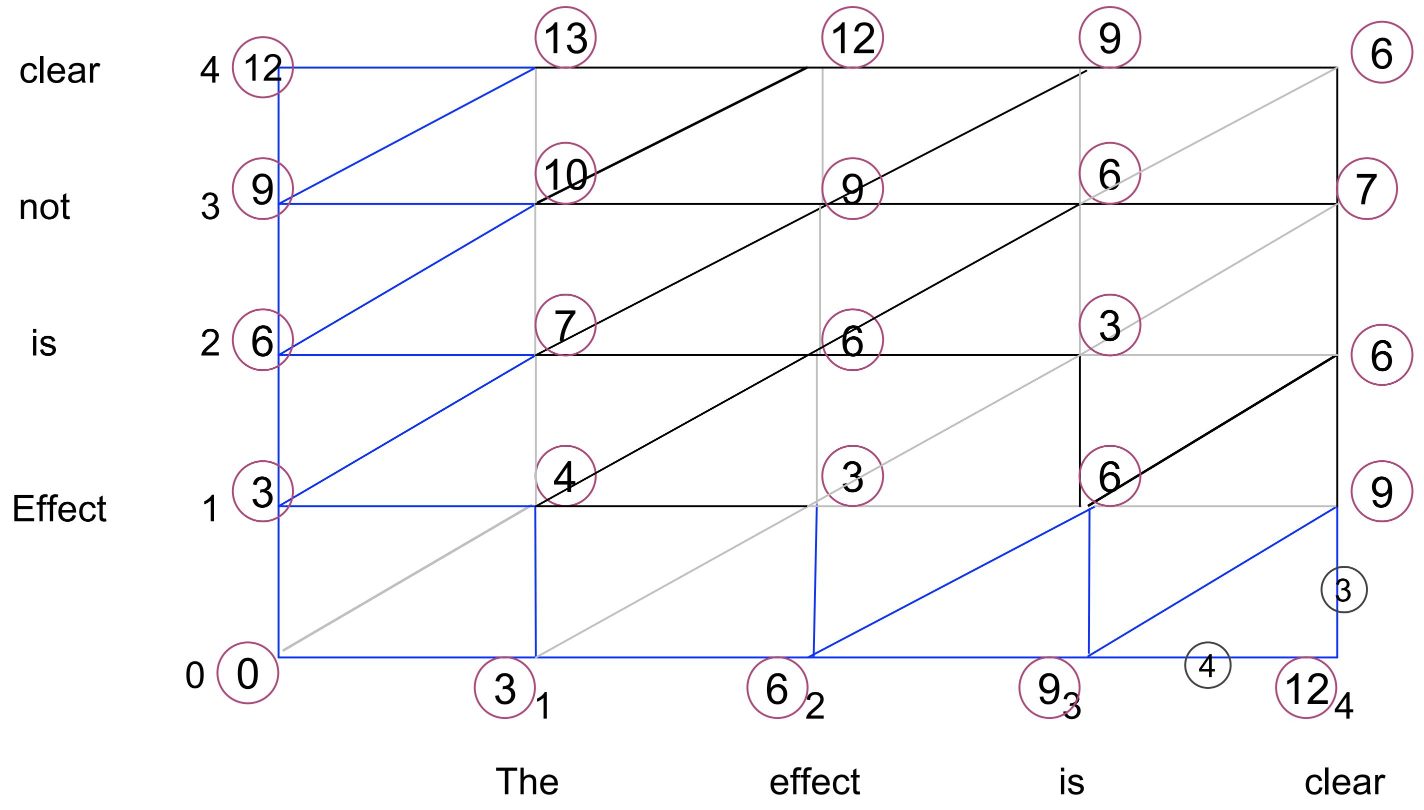


NTNU

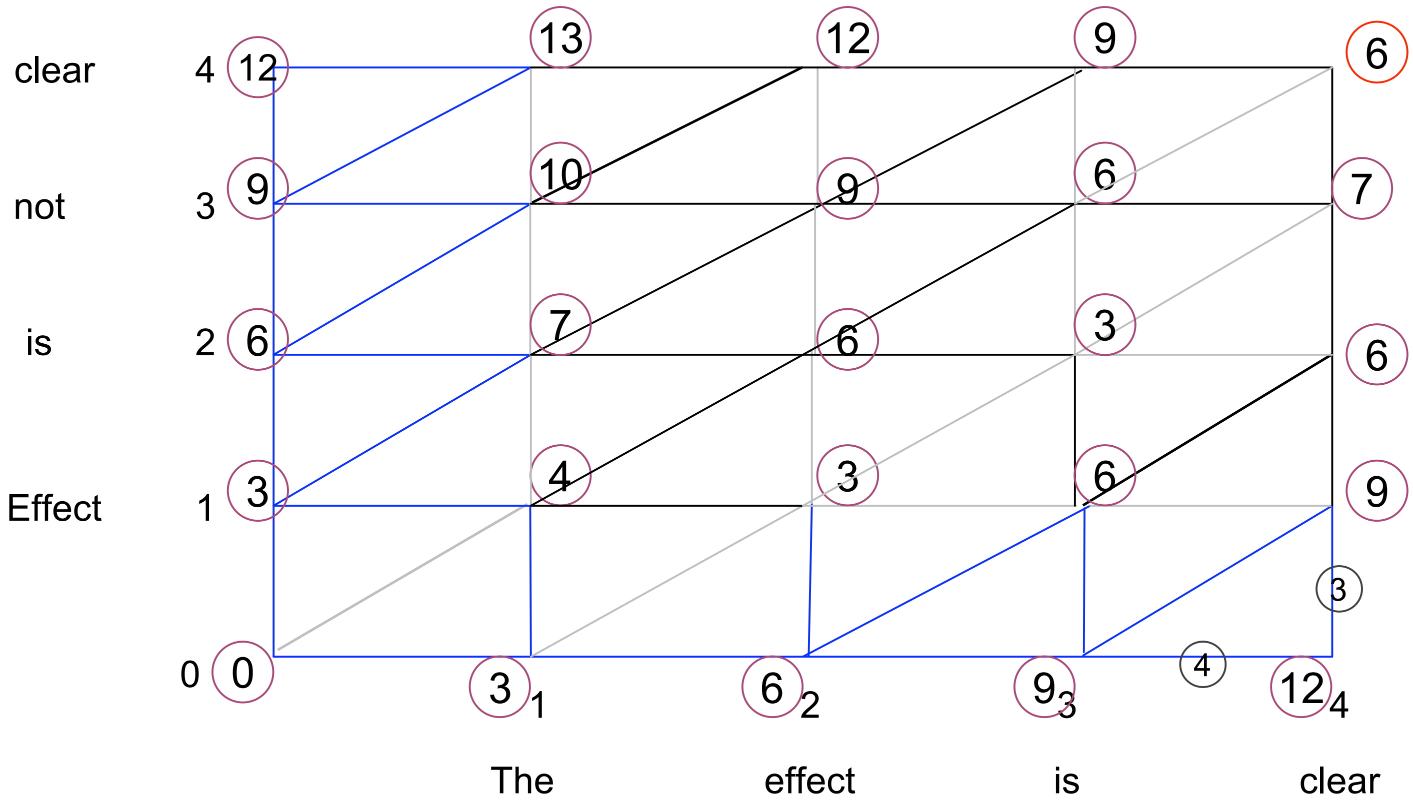
Dynamic programming example



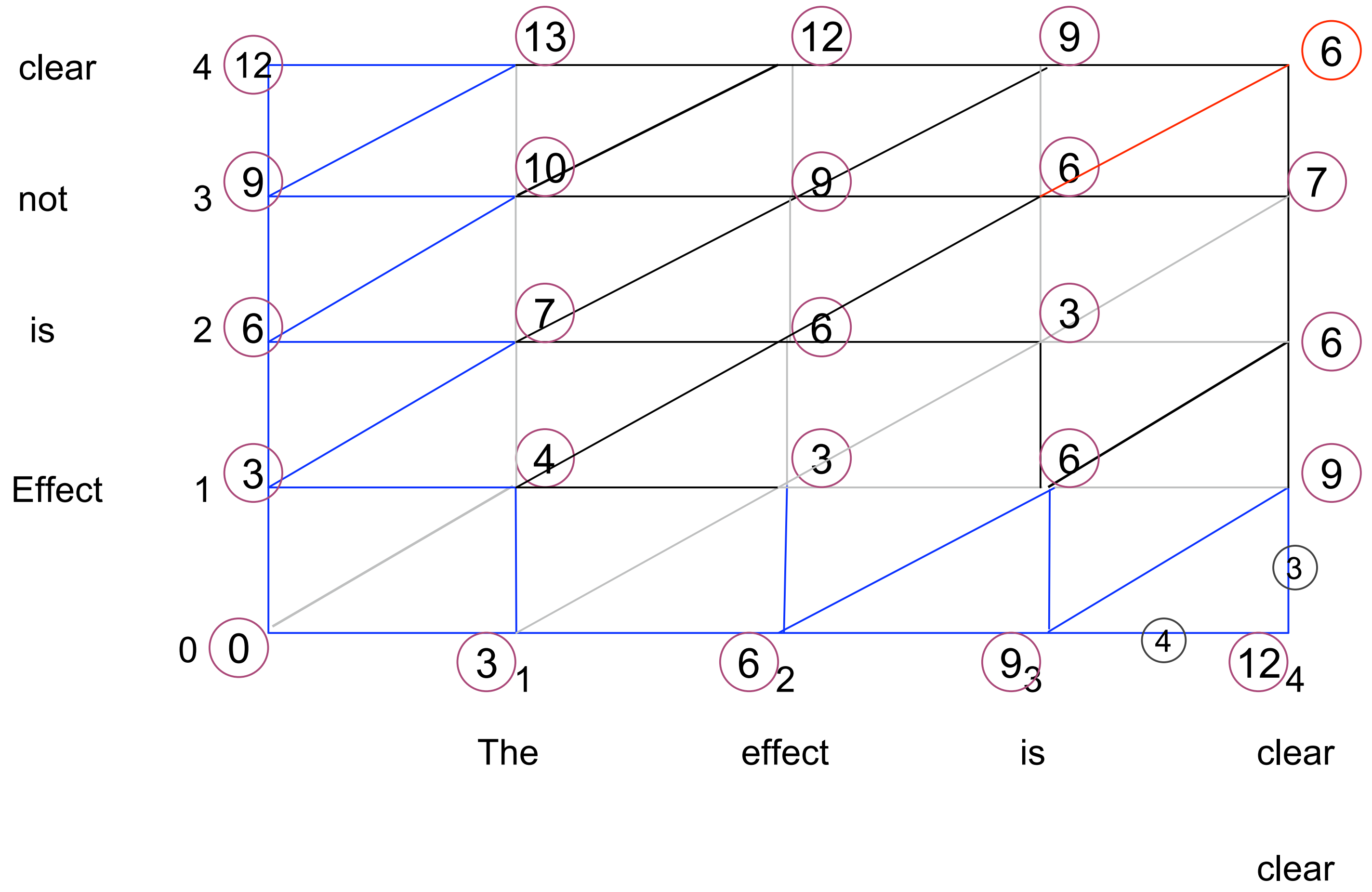
Dynamic programming example



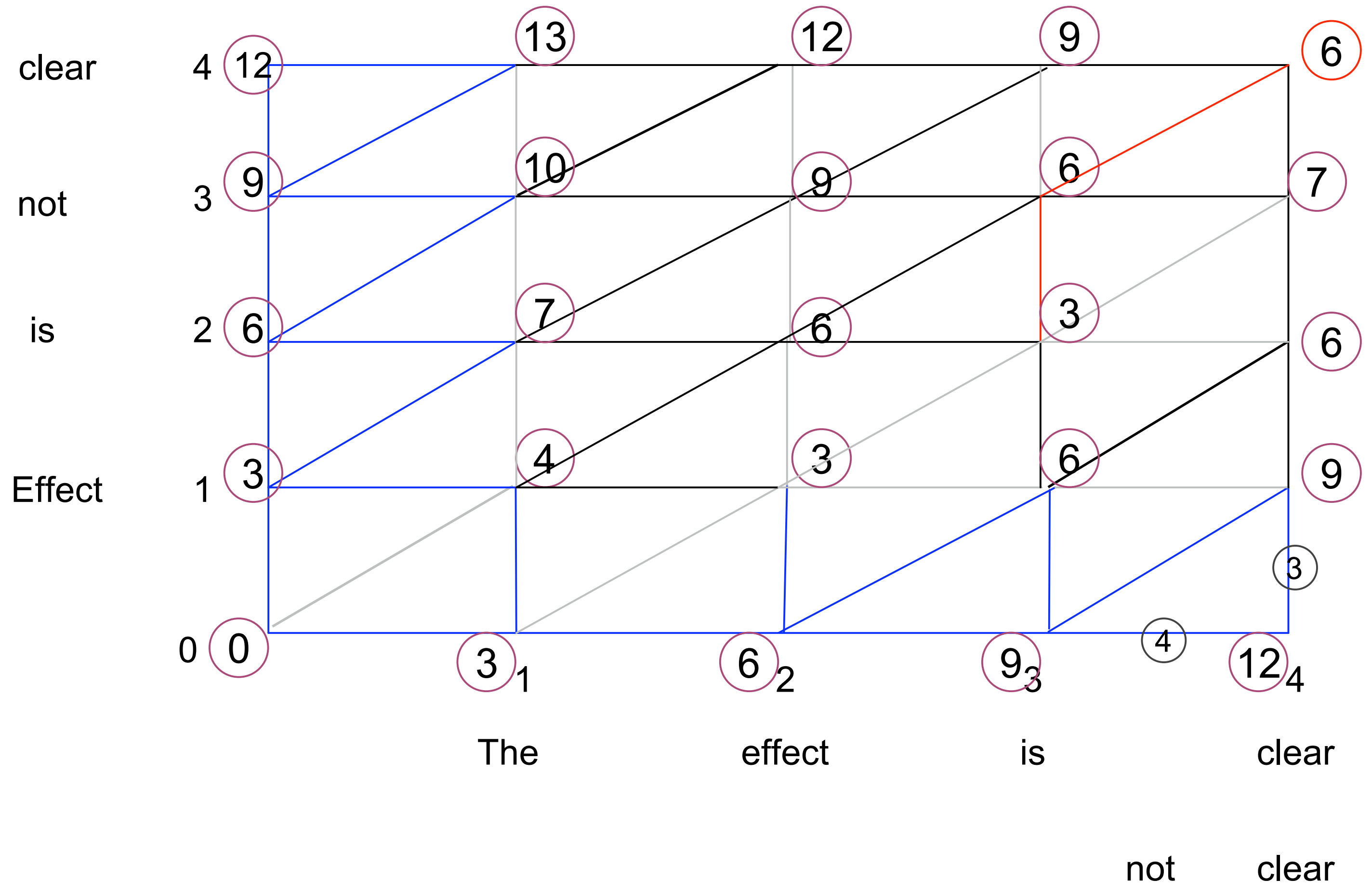
Dynamic programming example



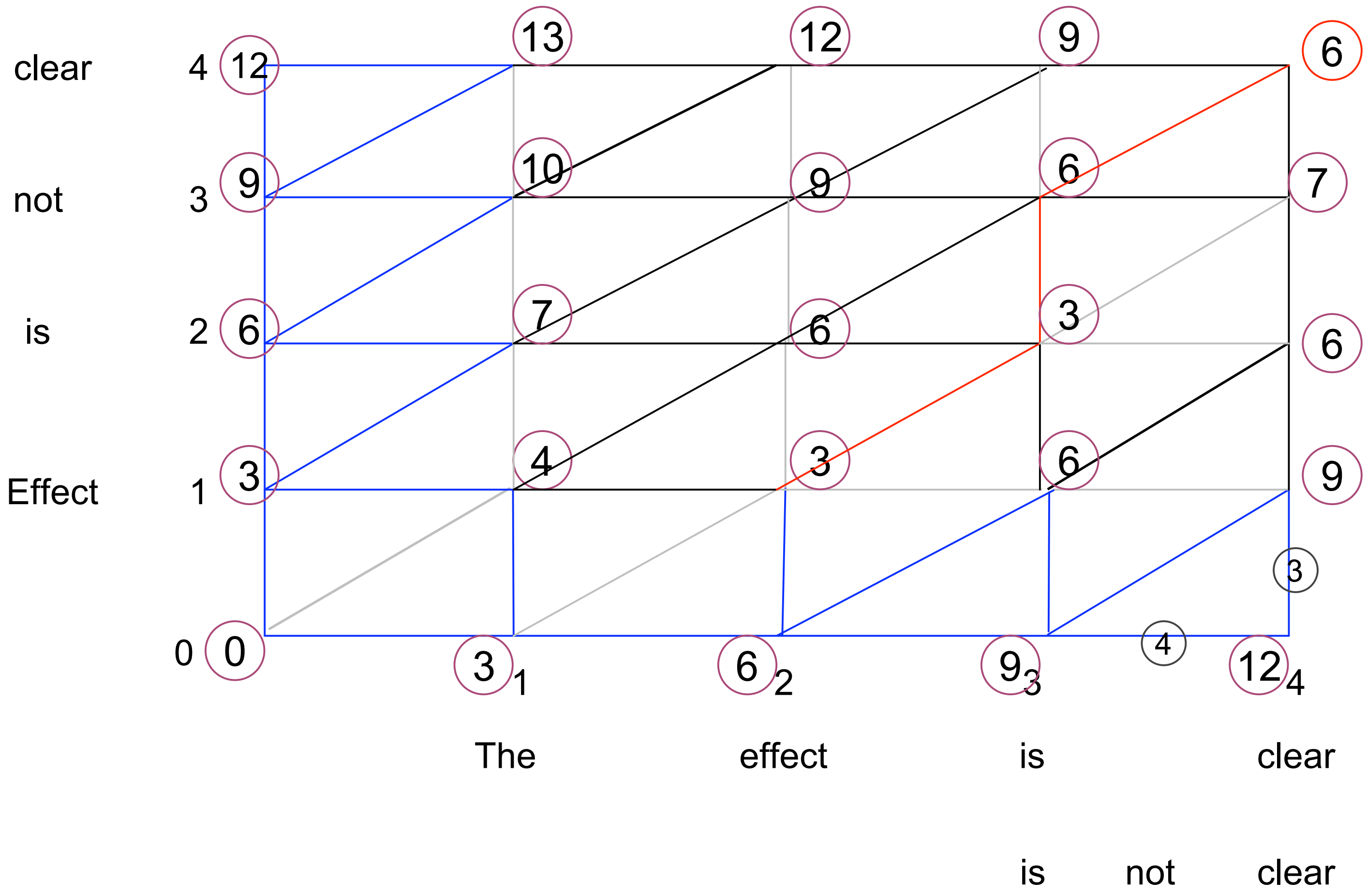
Dynamic programming example



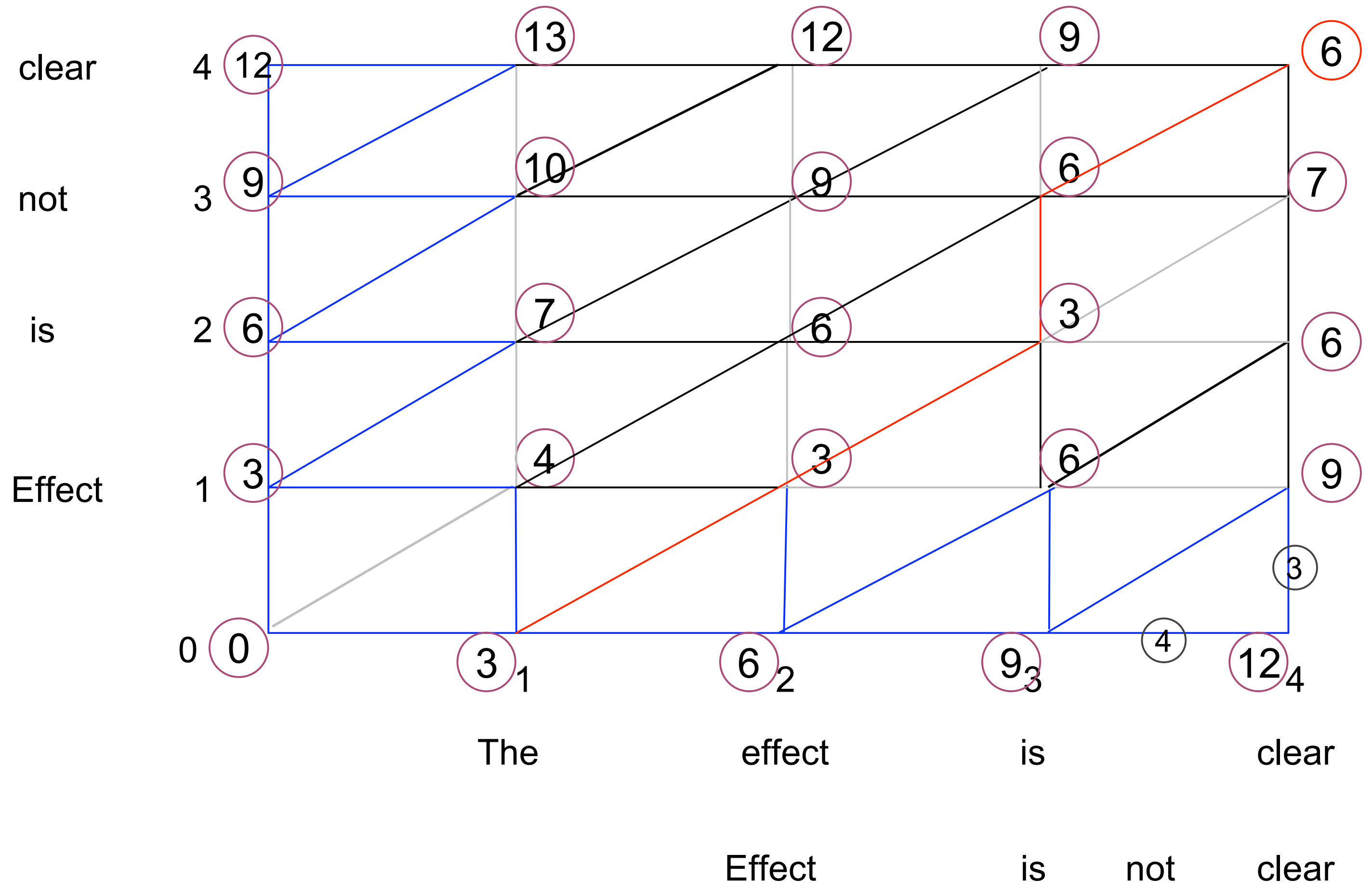
Dynamic programming example



Dynamic programming example



Dynamic programming example



Dynamic programming example