Efficient Semiring-Weighted Earley Parsing
Andreas Opedal, Ran Zmigrod, Tim Vieira, Ryan Cotterell and Jason Eisner

We present a fast version of Earley’s algorithm in the form of a unified deduction system that can be executed with good asymptotic complexity. We obtain further speedups by allowing the grammar to be presented in the form of a weighted finite-state automaton, which allows similar productions to share computation. We carefully generalize our methods to handle semiring-weighted grammars; providing a method to compute prefix weights for non-commutative semirings and showing how the same time and space complexity can be achieved for weighted parsing as for unweighted parsing (Eisner, 2023).

Why do we like Earley’s algorithm?
- Parses in \( \mathcal{O}(N^3) \) time for general grammars, in \( \mathcal{O}(N^2) \) time for unambiguous grammars and in \( \mathcal{O}(N) \) time for bounded-state grammars
- Incremental left-to-right sentence processing
- Can be extended to compute next-word probabilities (Stolcke, 1995)
  - Cognitive modeling (Hale, 2001)
  - Constrained generation of LLMs (Shin et al., 2021; Roy et al., 2022; Fang et al., 2023)

A faster Earley deduction system

<table>
<thead>
<tr>
<th>Earley</th>
<th>EarleyFast</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Domains</strong></td>
<td>( i, j, k \in {0, \ldots, N}, A, B \in \mathcal{N} \cup {S}, a \in \Sigma, \rho, \mu, v \in \langle \mathcal{N} \cup \Sigma \rangle^* )</td>
</tr>
</tbody>
</table>
| **Items** | \( [i, j, A \rightarrow \mu \cdot v] \quad [j, k, a] \rightarrow \rho \quad [i, j, A \rightarrow \mu \cdot \rho] 
            [i, j, A \rightarrow \rightarrow \cdot \rho] \quad [i, j, A \rightarrow \star \cdot \star] \) |
| **Axioms** | \( A \rightarrow \rho, \forall (A \rightarrow \rho) \in \mathcal{R} \quad A \rightarrow \rho, \forall (A \rightarrow \rho) \in \mathcal{R} 
            [k - 1, k, x_0], \forall k \in \{1, \ldots, N\} \quad [0, 0, S^* \rightarrow \star] \) |
| **Goal** | \( [0, 0, S \rightarrow \star \star] \) |
| **Pars:** | \( B \rightarrow \rho \rightarrow [j, j, B \rightarrow \star \star] \quad [i, j, A \rightarrow \mu \cdot B \cdot v] \) |
| **Rules:** | \( [i, j, A \rightarrow \mu \cdot v \cdot a] \quad [j, k, a] \quad [i, j, A \rightarrow \mu \cdot a \cdot v] 
            [i, k, A \rightarrow \mu \cdot a \cdot v] \) |
| **Comp:** | \( [i, j, A \rightarrow \mu \cdot B \cdot v] \quad [j, k, B \rightarrow \rho \cdot \star] \quad [i, k, A \rightarrow \mu \cdot B \cdot v] \) |

- 1-1 correspondence between proof trees and derivation trees
- From \( \mathcal{O}(N^3 |G| |R|) \) to \( \mathcal{O}(N^3 |G|) \)

EarleyFSA
- Encode weighted CFG as weighted FSA
- Production rules with similar right-hand sides can be partially merged (prefix and suffix sharing)
- \( \mathcal{O}(N^3 |M|) \), with \( |M| \leq |G| \)

Preliminaries for weighted parsing
- Each production rule is given a weight from the set of values of some (commutative) semiring \( \langle W, +, \cdot, 0, 1 \rangle \)
- The weight of a derivation tree is
  \[ w(T) \defeq \bigotimes_{(A \rightarrow \rho) \in T} w(A \rightarrow \rho) \]
- A weighted recognizer returns
  \[ Z_x \defeq w(S \xrightarrow{} x) \defeq \bigoplus_{T \in \mathcal{T}_x} w(T) \]
- A parser returns all derivation trees of \( x \)
- Prefix weights:
  \[ w(S \xrightarrow{}_{\star} y) \defeq \bigoplus_{a \in \Sigma^*} w(S \xrightarrow{}_{\star} yz) \]

Semiring-weighted deductive parsing
- The **incomplete inside weight** of an item \( V = [i, j, A \rightarrow \mu \cdot v] \) is the total weight of all its proofs
- Computed by \( \bigoplus \)-summing over the antecedent weights in all one-step proofs
- Requires computing **free sums**:
  - The **future inside weight** of \( V \) is the total weight of all ways to prove \( [i, j, A \rightarrow \mu \cdot v] \) from \( V \)
  - The **prefix outside weight** of \( V \) is the total weight of all ways to prove the goal item from \( [i, j, A \rightarrow \mu \cdot v] \)