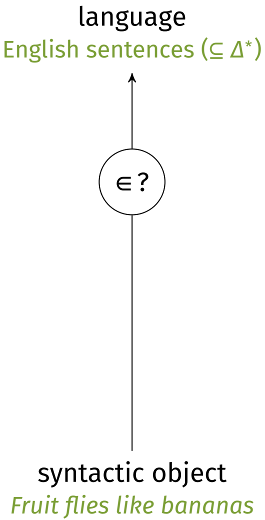
Weighted Parsing for Grammar-Based Language Models over Multioperator Monoids

WATA 2020/2021

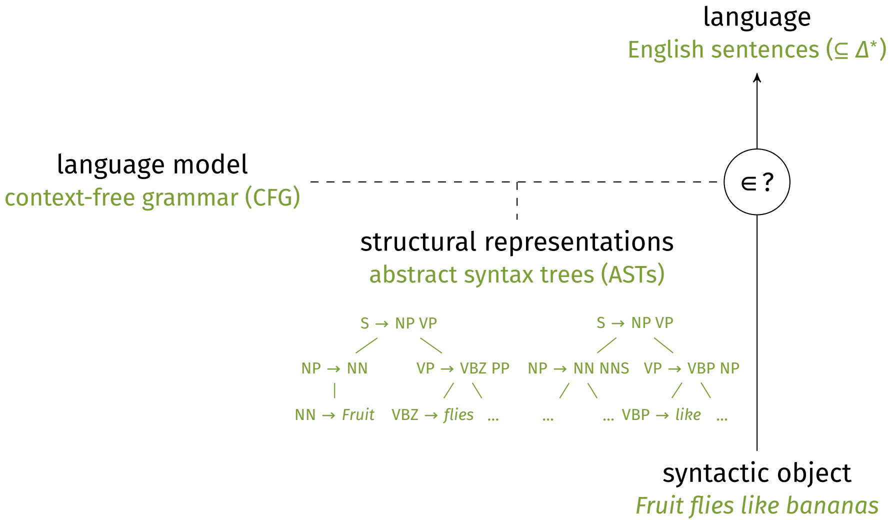
Richard Mörbitz   Heiko Vogler

2021-04-23

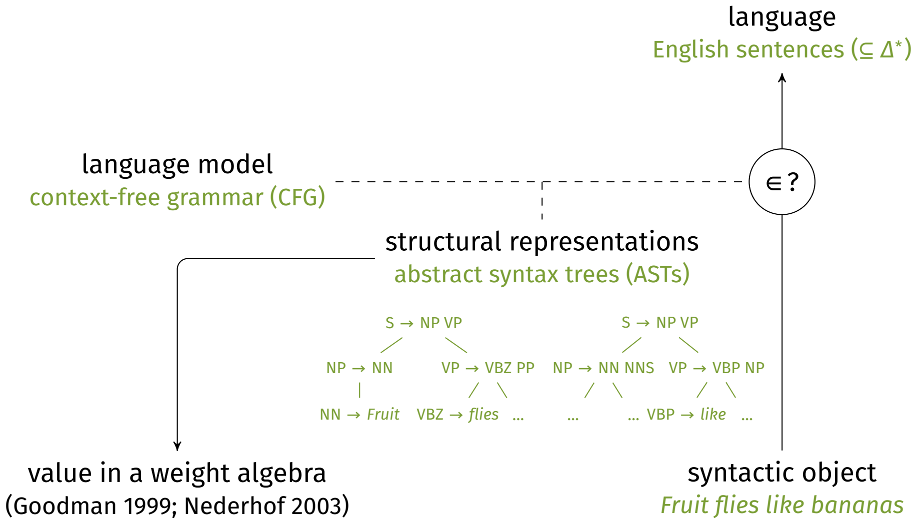
Weighted parsing problems



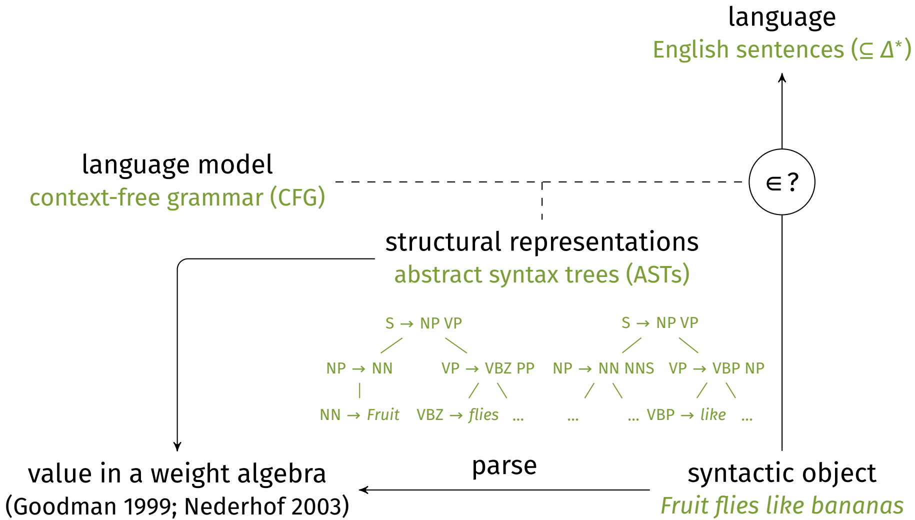
Weighted parsing problems



Weighted parsing problems



Weighted parsing problems



Outline

Our formalism

Weighted RTG-based language model

Our formalism

Weighted RTG-based language model

Regular tree grammar

$G = (N, \Sigma, A_0, R)$

Our formalism

Weighted RTG-based language model

Regular tree grammar
$G = (N, \Sigma, A_0, R)$

Language algebra
$(\mathcal{L}, \phi)$ is a Σ-algebra

Our formalism

Weighted RTG-based language model

- Regular tree grammar
  $G = (N, \Sigma, A_0, R)$
- Language algebra
  $(\mathcal{L}, \phi)$ is a Σ-algebra
- Multioperator monoid
  $(\mathbb{K}, \oplus, \mathbb{0}, \Omega)$

Our formalism

Weighted RTG-based language model

Regular tree grammar
$G = (N, \Sigma, A_0, R)$

Language algebra
$(\mathcal{L}, \phi)$ is a Σ-algebra

Multioperator monoid
$(\mathbb{K}, \oplus, \mathbb{0}, \Omega)$

from semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
with $\Omega = \{\odot^m_{\Bbbk} \mid m \in \mathbb{N}, \Bbbk \in \mathbb{K}\}$

Our formalism

Weighted RTG-based language model

Regular tree grammar
$G = (N, \Sigma, A_0, R)$

Language algebra
$(\mathcal{L}, \phi)$ is a Σ-algebra

Multioperator monoid
$(\mathbb{K}, \oplus, \mathbb{0}, \Omega)$

from semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
with $\Omega = \{\odot_{\Bbbk}^{m} \mid m \in \mathbb{N}, \Bbbk \in \mathbb{K}\}$

Our formalism

Weighted RTG-based language model

Regular tree grammar
$G = (N, \Sigma, A_0, R)$

Language algebra
$(\mathcal{L}, \phi)$ is a Σ-algebra

Multioperator monoid
$(\mathbb{K}, \oplus, \mathbb{0}, \Omega)$

wt

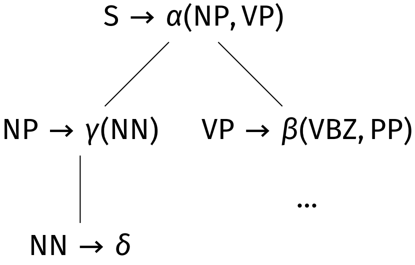
from semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
with $\Omega = \{\odot_{\Bbbk}^{m} \mid m \in \mathbb{N}, \Bbbk \in \mathbb{K}\}$

Weighted *RTG-based* language models

Tuple $G = (N, \Sigma, A_0, R)$

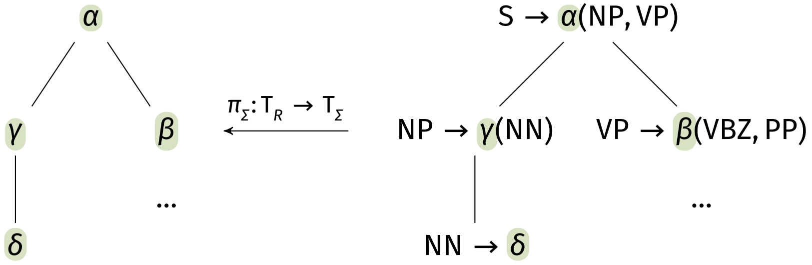
Example rules:

$$\begin{aligned} \mathrm{S} &\to \alpha(\mathrm{NP}, \mathrm{VP}) \\ \mathrm{VP} &\to \beta(\mathrm{VBZ}, \mathrm{PP}) \\ \mathrm{NP} &\to \gamma(\mathrm{NN}) \\ \mathrm{NN} &\to \delta \\ &\dots \end{aligned}$$



abstract syntax tree $d \in \mathrm{AST}(G)$

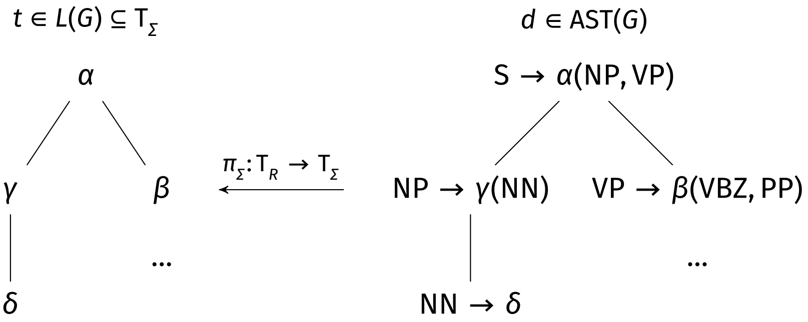
Weighted RTG-based *language models*



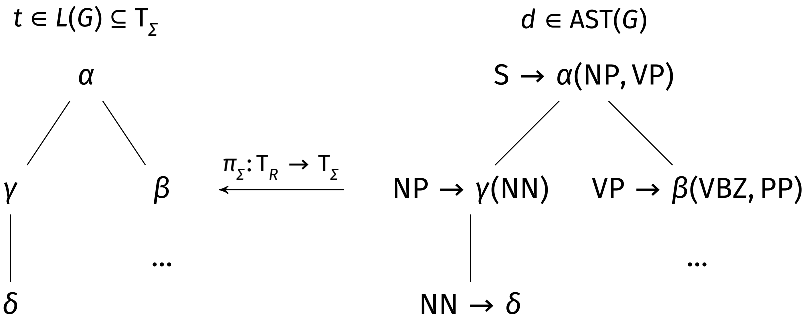
$t \in L(G) \subseteq \mathrm{T}_\Sigma$

abstract syntax tree $d \in \mathrm{AST}(G)$

Weighted RTG-based *language models*

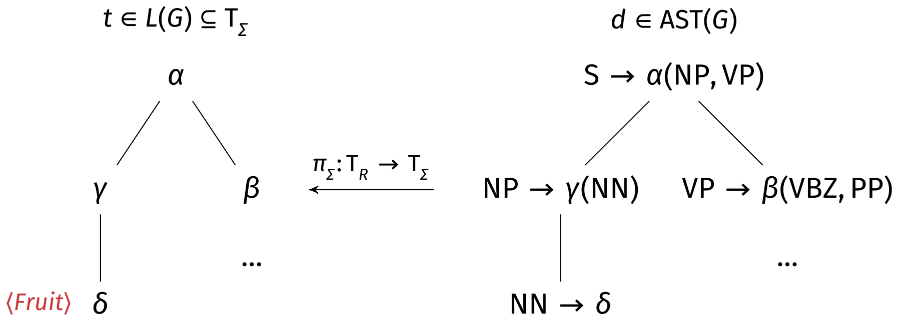


Weighted RTG-based *language models*



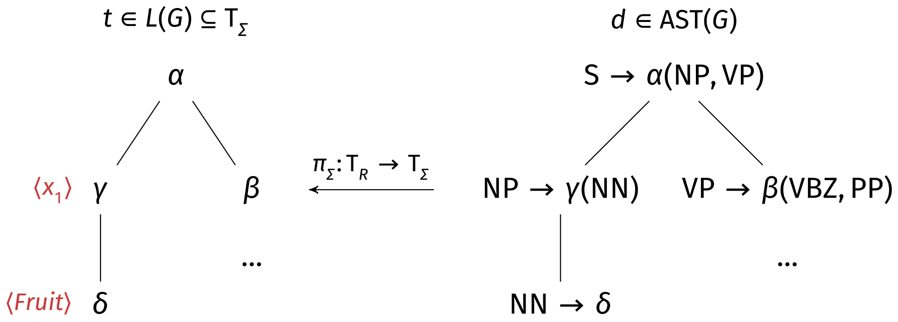
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{L} = \Delta^*$

Weighted RTG-based *language models*



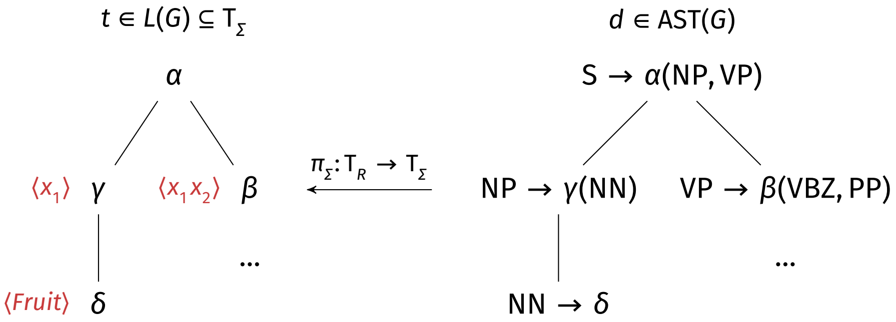
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$

Weighted RTG-based *language models*



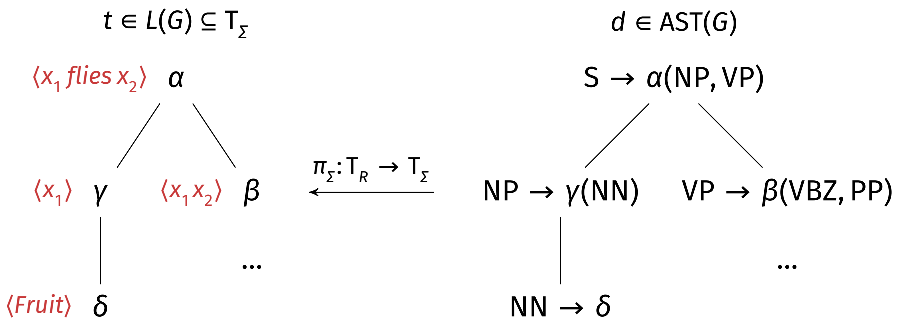
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$

Weighted RTG-based *language models*



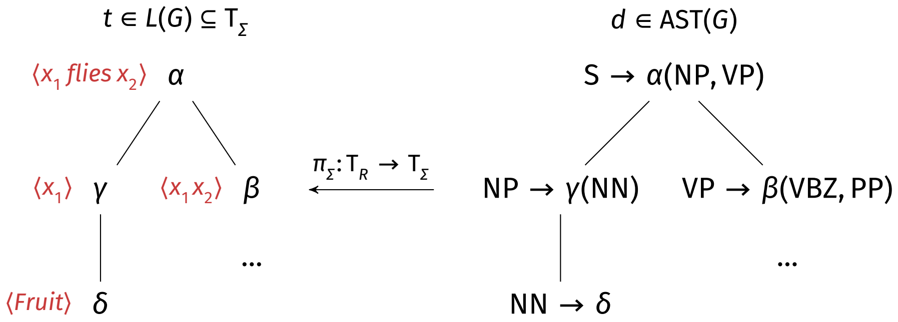
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{L} = \Delta^*$

Weighted RTG-based *language models*



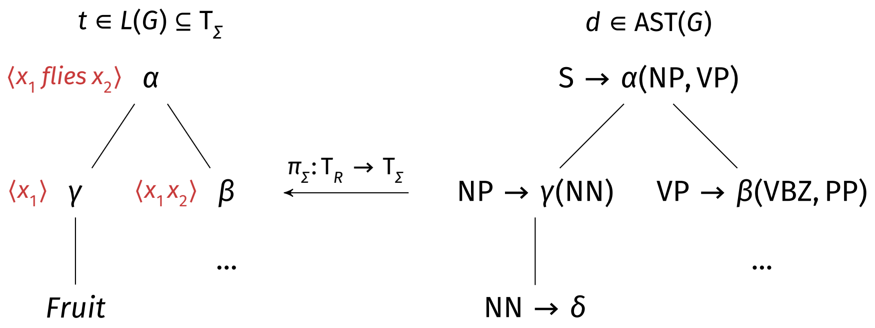
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$

Weighted RTG-based *language models*



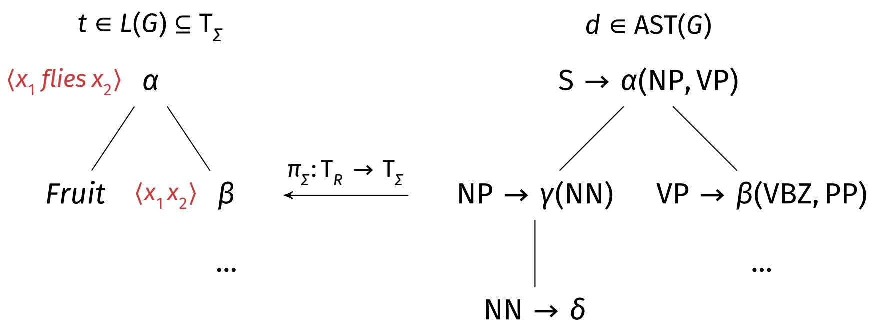
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$
- $(.)_{\Delta^*}: T_\Sigma$ (terms) $\to \Delta^*$ (syntactic objects)

Weighted RTG-based *language models*



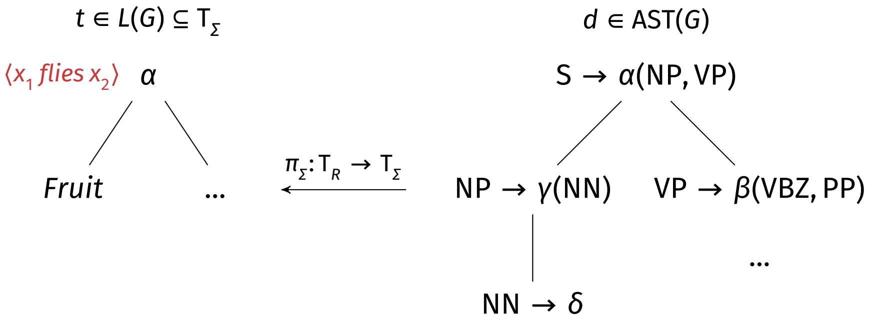
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$
- $(.)_{\Delta^*} : \mathrm{T}_\Sigma$ (terms) $\to \Delta^*$ (syntactic objects)

Weighted RTG-based *language models*



- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$
- $(.)_{\Delta^*}: \mathrm{T}_\Sigma$ (terms) $\to \Delta^*$ (syntactic objects)

Weighted RTG-based *language models*



- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$
- $(.)_{\Delta^*} : \mathrm{T}_\Sigma$ (terms) $\to \Delta^*$ (syntactic objects)

Weighted RTG-based *language models*

$t \in L(G) \subseteq \mathrm{T}_\Sigma$

$\langle \textit{Fruit flies} \ldots \rangle$

$d \in \mathrm{AST}(G)$

$\mathrm{S} \to \alpha(\mathrm{NP}, \mathrm{VP})$

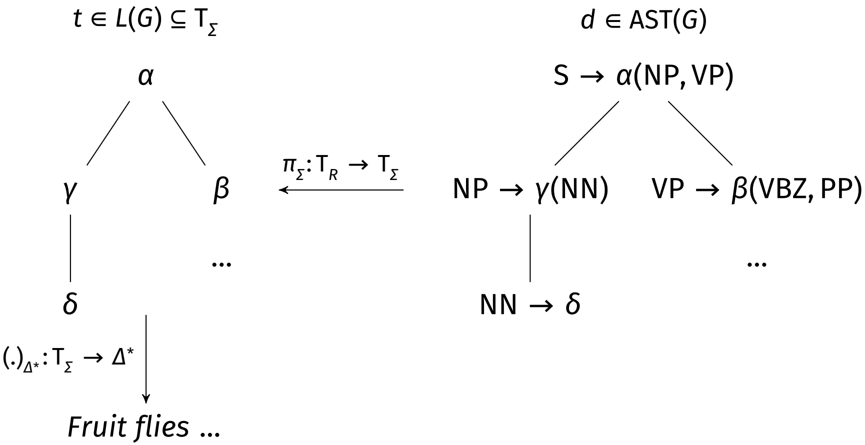
$\pi_\Sigma : \mathrm{T}_R \to \mathrm{T}_\Sigma$ ←

$\mathrm{NP} \to \gamma(\mathrm{NN})$ $\quad$ $\mathrm{VP} \to \beta(\mathrm{VBZ}, \mathrm{PP})$

$\mathrm{NN} \to \delta$ $\quad$ …

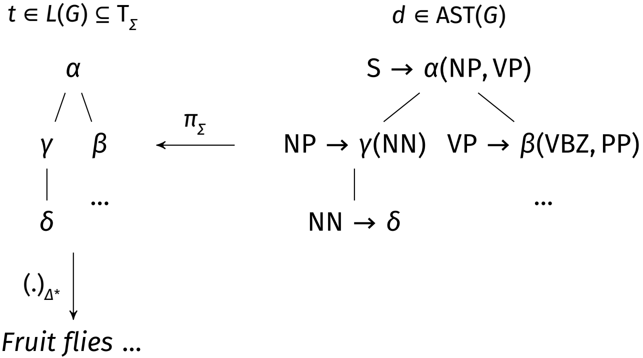
- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$
- $(.)_{\Delta^*} : \mathrm{T}_\Sigma$ (terms) $\to \Delta^*$ (syntactic objects)

Weighted RTG-based *language models*

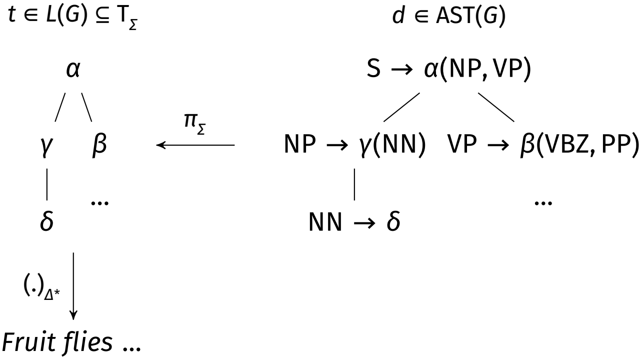


- ϕ: interpret Σ as operations on the set of syntactic objects $\mathcal{C} = \Delta^*$
- $(.)_{\Delta^*} : T_\Sigma$ (terms) $\to \Delta^*$ (syntactic objects)

Weighted RTG-based language models

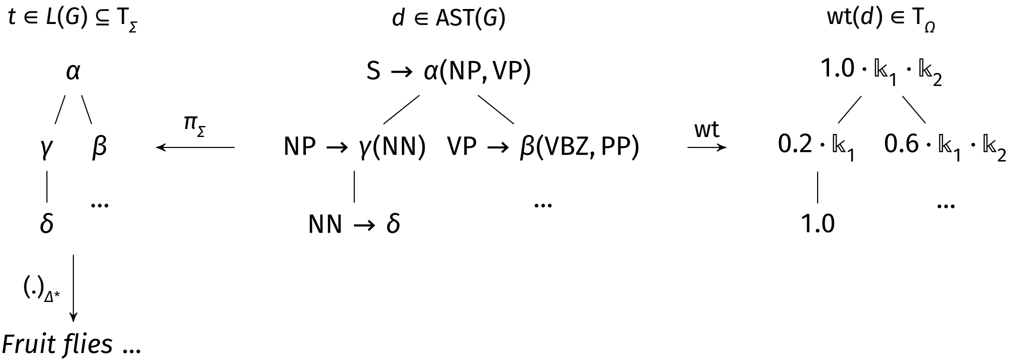


Weighted RTG-based language models



- wt: R (set of rules) $\to \Omega$ (set of operations) $(\mathbb{R}_0^1, \max, 0, \Omega_{\mathrm{mul}})$

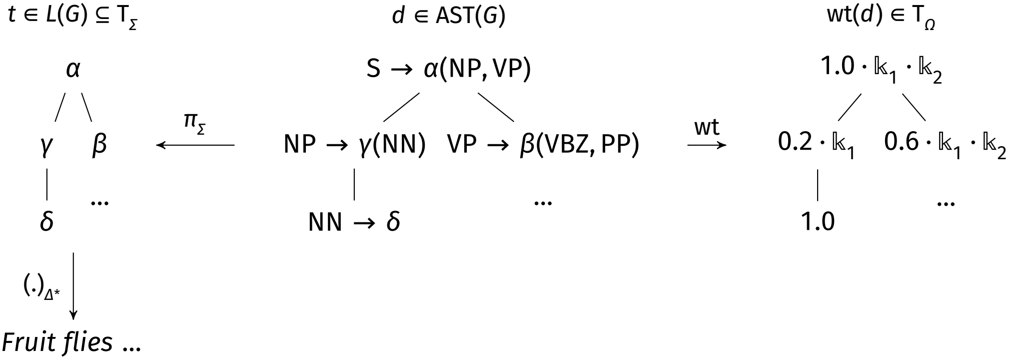
Weighted RTG-based language models



- wt: R (set of rules) $\to$ Ω (set of operations)

$(\mathbb{R}_0^1, \max, 0, \Omega_{mul})$

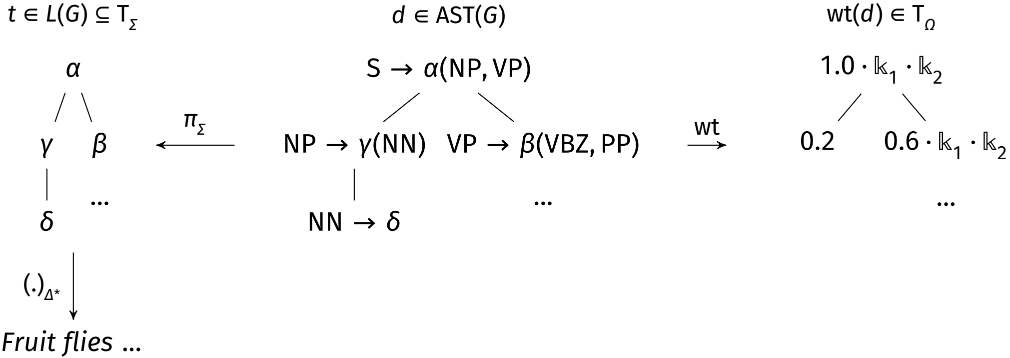
Weighted RTG-based language models



- wt: R (set of rules) $\to$ Ω (set of operations)
- $(.)_{\mathbb{R}_1^0}$: T_Ω (terms) $\to$ $\mathbb{R}_0^1$ (weight algebra)

$(\mathbb{R}_0^1, \max, 0, \Omega_{\mathrm{mul}})$

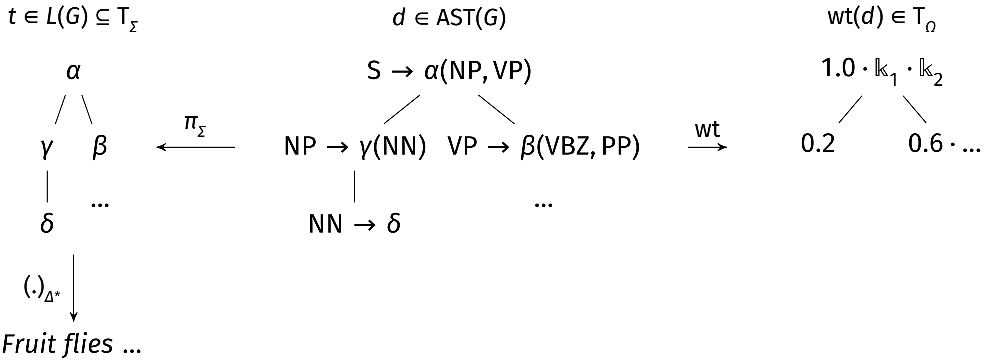
Weighted RTG-based language models



- wt: R (set of rules) $\to$ Ω (set of operations)
- $(.)_{\mathbb{R}_1^0}$: T_Ω (terms) $\to$ $\mathbb{R}_0^1$ (weight algebra)

$(\mathbb{R}_0^1, \max, 0, \Omega_{mul})$

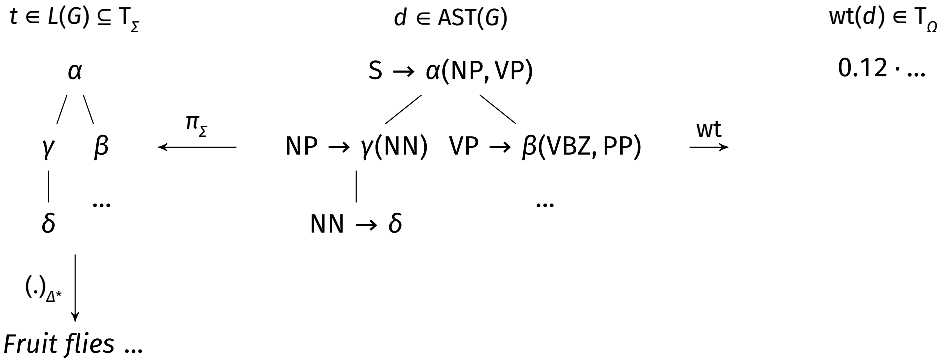
Weighted RTG-based language models



- wt: R (set of rules) $\to \Omega$ (set of operations)
- $(.)_{\mathbb{R}^0_1}: T_\Omega$ (terms) $\to \mathbb{R}^1_0$ (weight algebra)

$(\mathbb{R}^1_0, \max, 0, \Omega_{mul})$

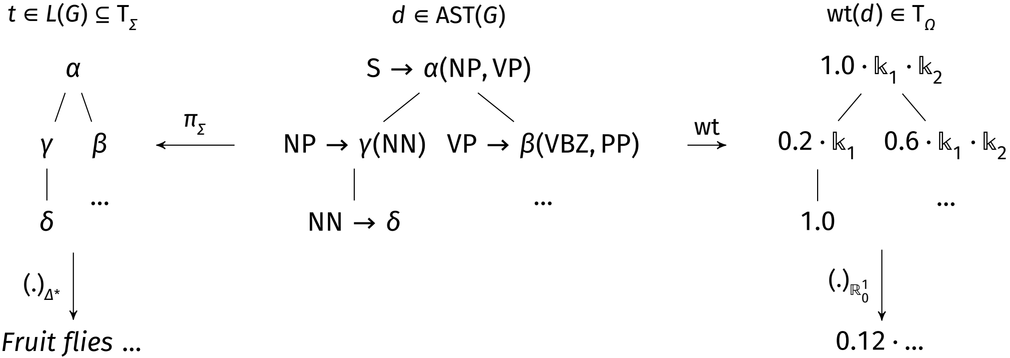
Weighted RTG-based language models



- wt: R (set of rules) → Ω (set of operations)
- $(.)_{\mathbb{R}_1^0}$: T_Ω (terms) → $\mathbb{R}_0^1$ (weight algebra)

$(\mathbb{R}_0^1, \max, 0, \Omega_{mul})$

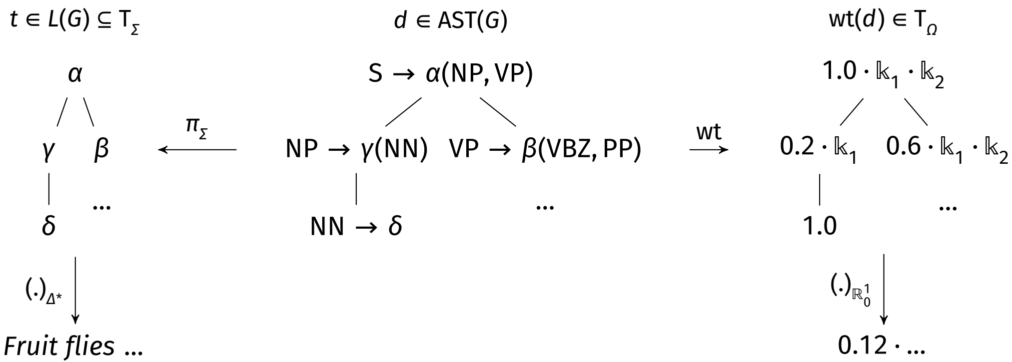
Weighted RTG-based language models



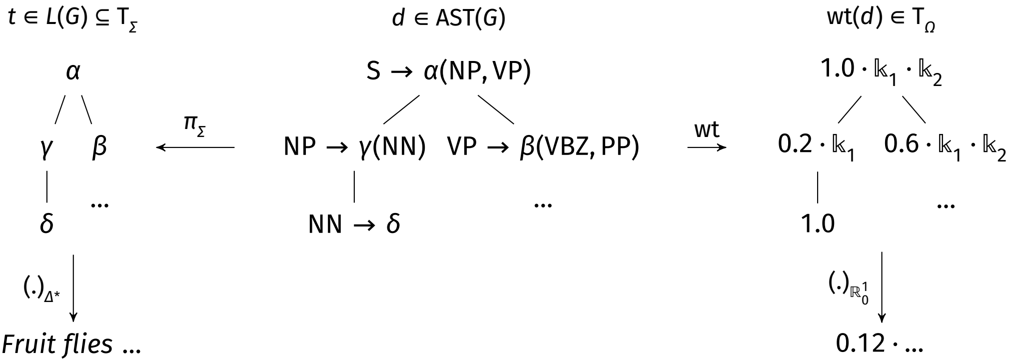
- wt: R (set of rules) $\to \Omega$ (set of operations)
- $(.)_{\mathbb{R}^0_1}$: T_Ω (terms) $\to \mathbb{R}^1_0$ (weight algebra)

$(\mathbb{R}^1_0, \max, 0, \Omega_{mul})$

Weighted RTG-based language models



Weighted RTG-based language models

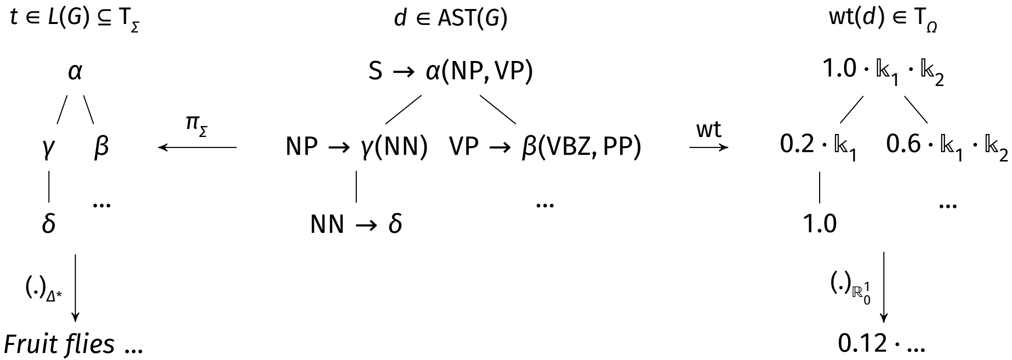


Definition (weighted RTG-based language model)

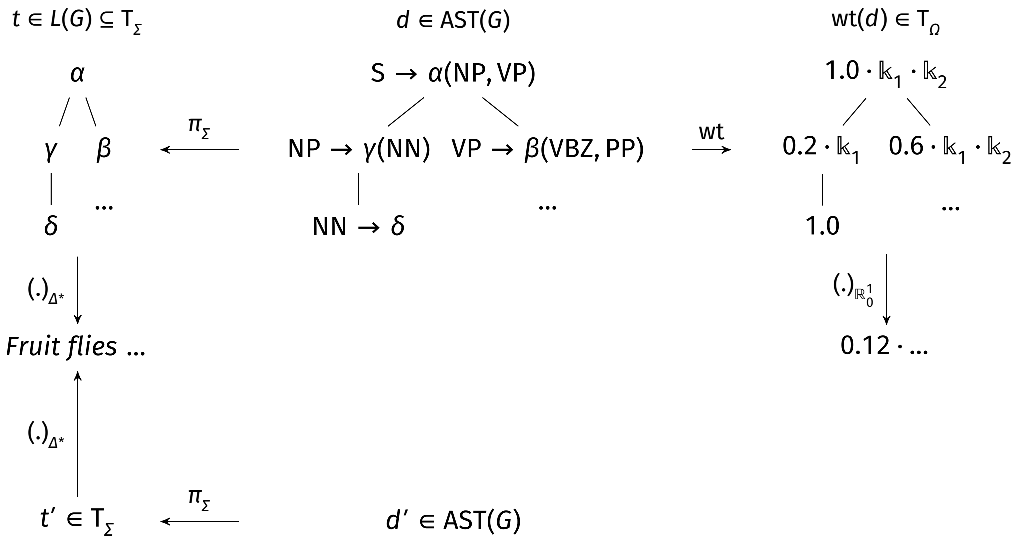
A wRTG-LM is a tuple $\Big(\underbrace{(G = (N, \Sigma, A_0, R))}_{\text{RTG}}, \underbrace{(\mathcal{L}, \phi)}_{\text{language algebra}}), \underbrace{(\mathbb{K}, \oplus, \mathbb{0}, \Omega)}_{\text{M-monoid}}, \underbrace{\mathrm{wt}}_{\mathrm{wt}\colon R \rightarrow \Omega}\Big)$.

Outline

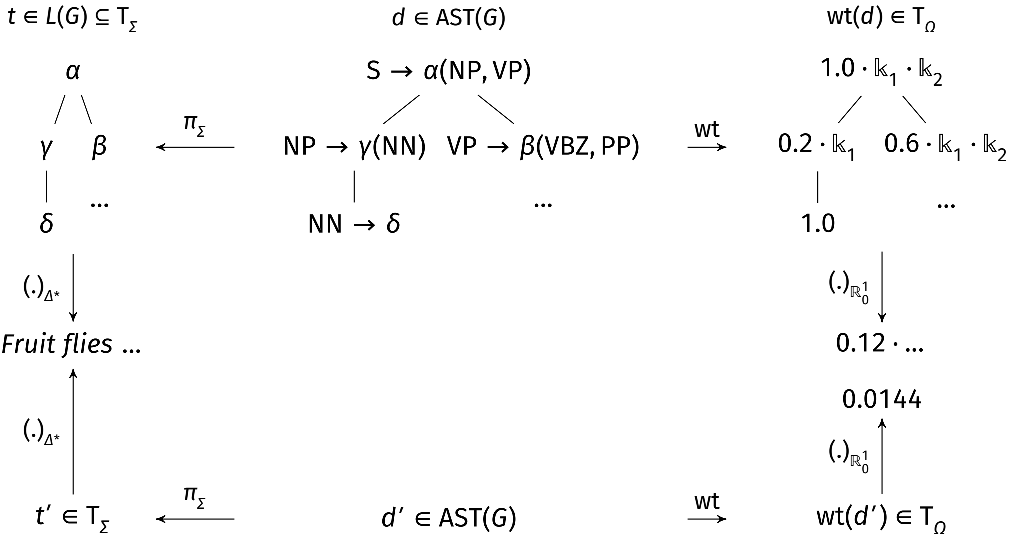
The M-monoid parsing problem



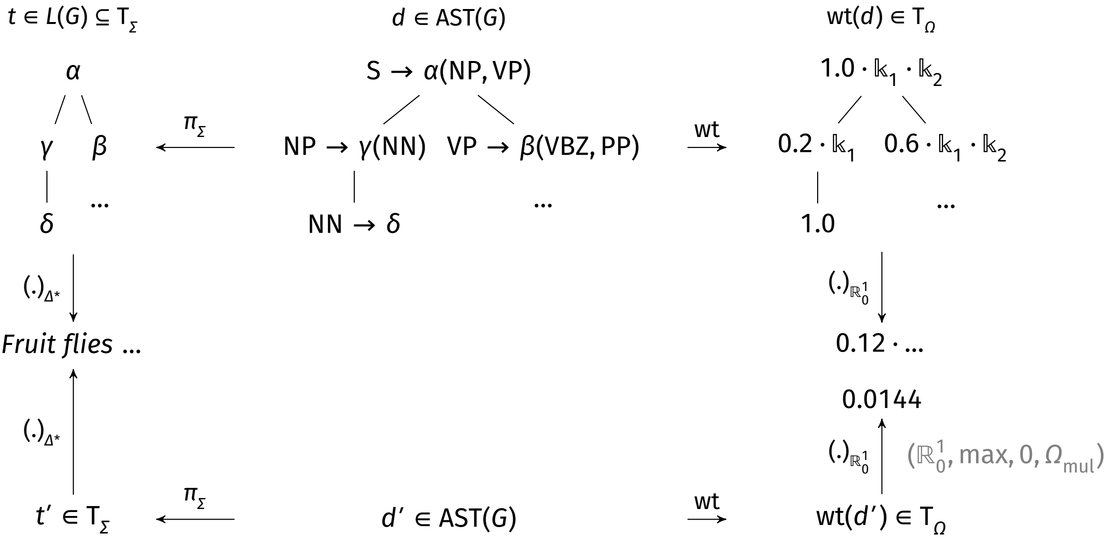
The M-monoid parsing problem



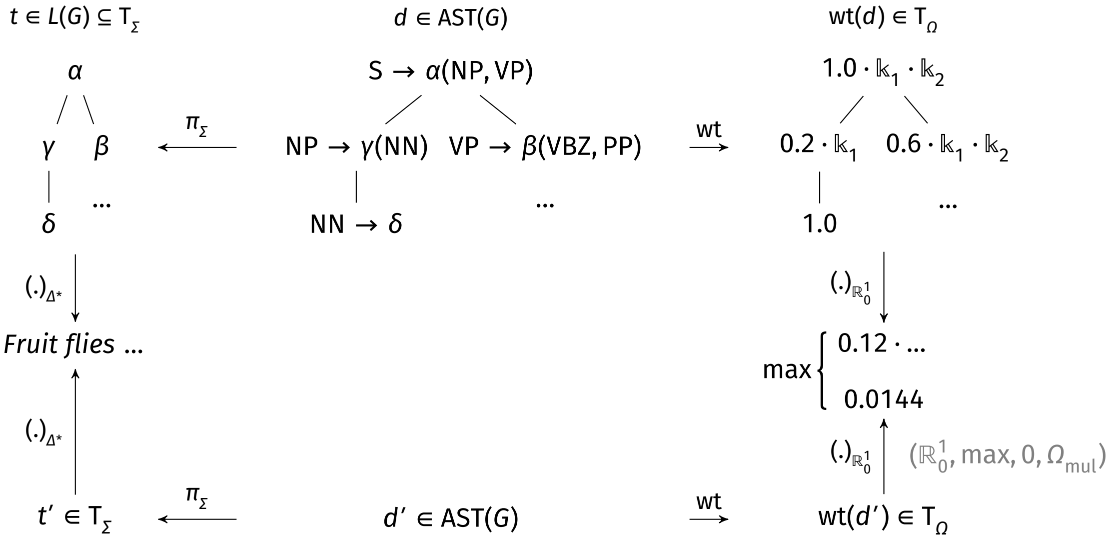
The M-monoid parsing problem



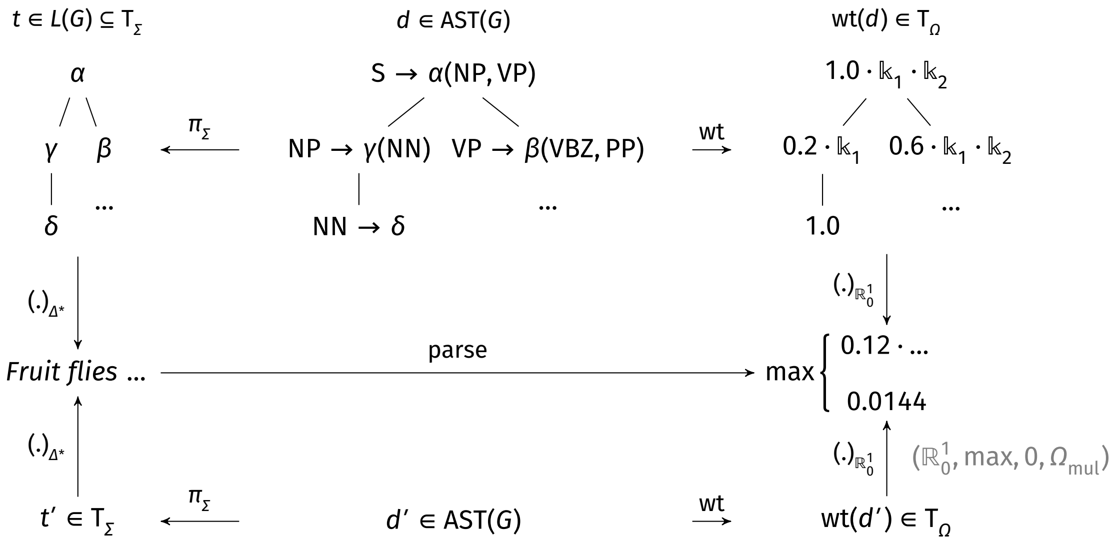
The M-monoid parsing problem



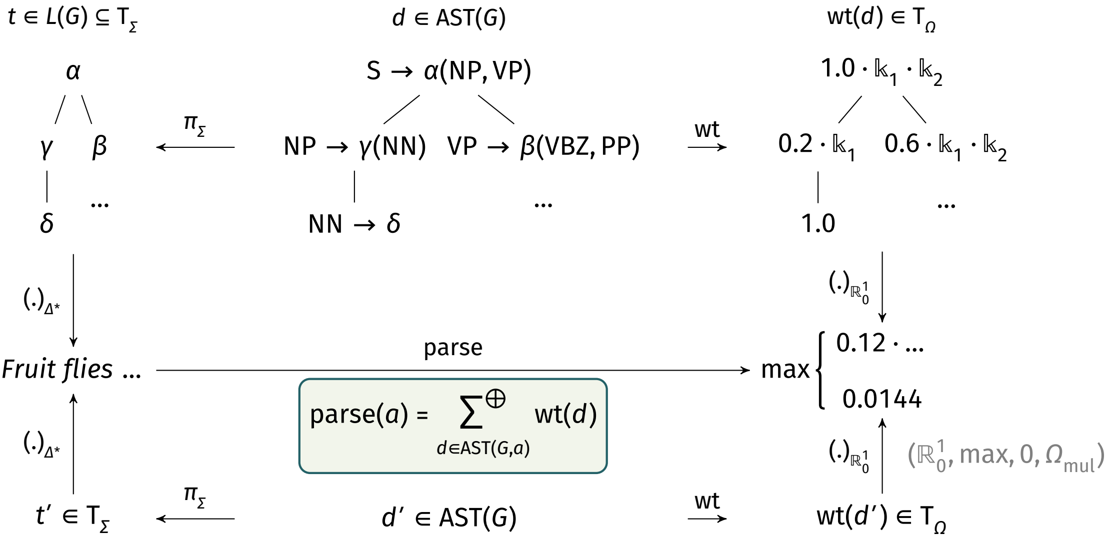
The M-monoid parsing problem



The M-monoid parsing problem



The M-monoid parsing problem



Outline

The M-monoid parsing algorithm

Two-phase pipeline (Goodman 1999; Nederhof 2003)

The M-monoid parsing algorithm

Two-phase pipeline (Goodman 1999; Nederhof 2003)

- wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$
- $a \in \mathcal{L}$

$$\mathrm{parse}(a) = \sum^{\oplus}_{d \in \mathrm{AST}(G,a)} \mathrm{wt}(d)$$

The M-monoid parsing algorithm

Two-phase pipeline (Goodman 1999; Nederhof 2003)

- wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$
- $a \in \mathcal{L}$

→ canonical weighted deduction system

$$\mathrm{parse}(a) = \sum_{d \in \mathrm{AST}(G,a)}^{\oplus} \mathrm{wt}(d)$$

The M-monoid parsing algorithm

Two-phase pipeline (Goodman 1999; Nederhof 2003)

- wRTG-LM $((G, \mathcal{L}), \mathbb{K}, \mathrm{wt})$
- $a \in \mathcal{L}$

$\longrightarrow$ canonical weighted deduction system $\longrightarrow$ wRTG-LM $((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}')$

$$\mathrm{parse}(a) = \sum^{\oplus}_{d \in \mathrm{AST}(G,a)} \mathrm{wt}(d)$$

The M-monoid parsing algorithm

Two-phase pipeline (Goodman 1999; Nederhof 2003)

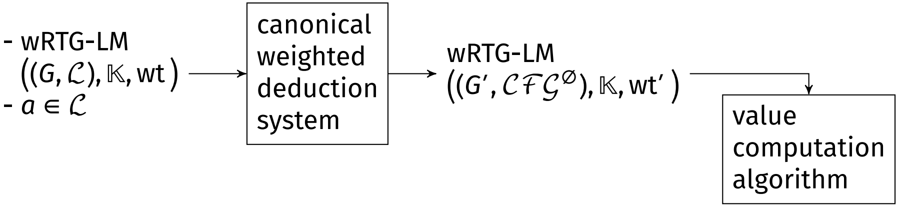
- wRTG-LM $((G, \mathcal{L}), \mathbb{K}, \mathrm{wt})$
- $a \in \mathcal{L}$

$\longrightarrow$ canonical weighted deduction system $\longrightarrow$ wRTG-LM $((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}')$

$$\mathrm{parse}(a) = \sum^{\oplus}_{d \in \mathrm{AST}(G,a)} \mathrm{wt}(d) \stackrel{?}{=} \sum^{\oplus}_{d \in \mathrm{AST}(G')} \mathrm{wt}'(d)$$

The M-monoid parsing algorithm

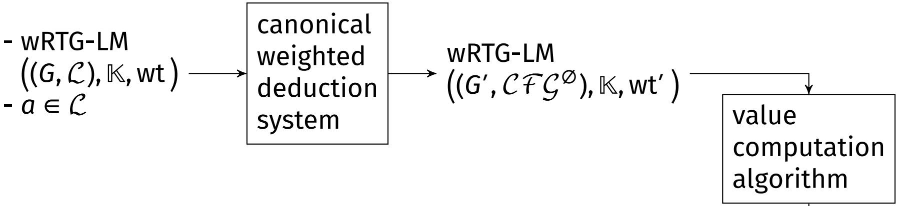
Two-phase pipeline (Goodman 1999; Nederhof 2003)



$$\mathrm{parse}(a) = \sum^{\oplus}_{d \in \mathrm{AST}(G,a)} \mathrm{wt}(d) \overset{?}{=} \sum^{\oplus}_{d \in \mathrm{AST}(G')} \mathrm{wt}'(d)$$

The M-monoid parsing algorithm

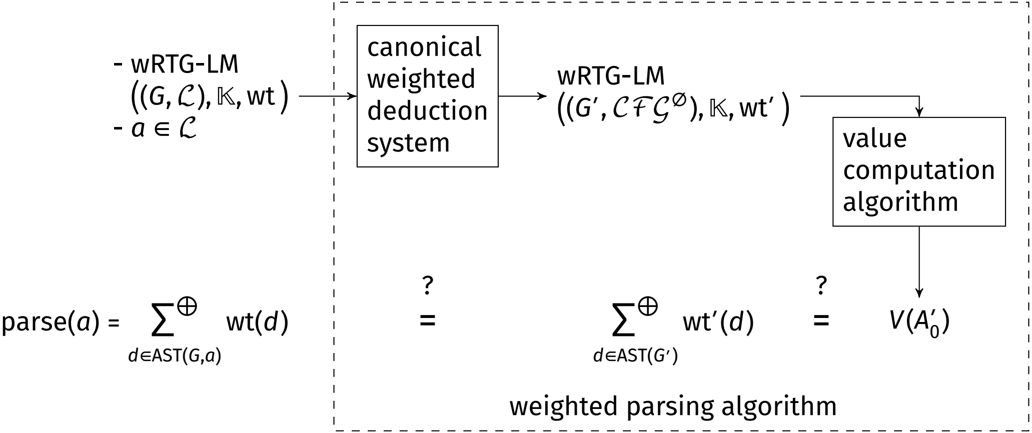
Two-phase pipeline (Goodman 1999; Nederhof 2003)



$$\mathrm{parse}(a) = \sum^{\oplus}_{d \in \mathrm{AST}(G,a)} \mathrm{wt}(d) \overset{?}{=} \sum^{\oplus}_{d \in \mathrm{AST}(G')} \mathrm{wt}'(d) \overset{?}{=} V(A'_0)$$

The M-monoid parsing algorithm

Two-phase pipeline (Goodman 1999; Nederhof 2003)



The canonical weighted deduction system

- wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$
- $a \in \mathcal{L}$

$\xrightarrow{\text{cwds}}$ wRTG-LM $\big((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}'\big)$

Parsing as deduction (Shieber, Schabes, and Pereira 1995)

$$\frac{[A_1, a_1] \;\dots\; [A_m, a_m]}{[A, a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule in } G \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \phi(\sigma)(a_1, \dots, a_m) \end{cases}$$

The canonical weighted deduction system

- wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$
- $a \in \mathcal{L}$

$\xrightarrow{\text{cwds}}$ wRTG-LM $\big((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}'\big)$

Parsing as deduction (Shieber, Schabes, and Pereira 1995)

$$\frac{[A_1, a_1] \ \dots \ [A_m, a_m]}{[A, a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule in } G \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \phi(\sigma)(a_1, \dots, a_m) \end{cases}$$

The canonical weighted deduction system

- wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$
- $a \in \mathcal{L}$

$\xrightarrow{\text{cwds}}$ wRTG-LM $\big((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}'\big)$

Parsing as deduction (Shieber, Schabes, and Pereira 1995)

$$\frac{[A_1, a_1] \;\dots\; [A_m, a_m]}{[A, a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule in } G \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \phi(\sigma)(a_1, \dots, a_m) \end{cases}$$

The canonical weighted deduction system

- wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$ $\xrightarrow{\text{cwds}}$ wRTG-LM $\big((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}'\big)$
- $a \in \mathcal{L}$

Parsing as deduction (Shieber, Schabes, and Pereira 1995)

$$\frac{[A_1, a_1] \ \dots \ [A_m, a_m]}{[A, a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule in } G \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \phi(\sigma)(a_1, \dots, a_m) \end{cases}$$

The canonical weighted deduction system

- wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big) \xrightarrow{\text{cwds}}$ wRTG-LM $\big((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}'\big)$
- $a \in \mathcal{L}$

Parsing as deduction (Shieber, Schabes, and Pereira 1995)

$$\frac{[A_1, a_1] \ \dots \ [A_m, a_m]}{[A, a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule in } G \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \phi(\sigma)(a_1, \dots, a_m) \end{cases}$$

Rule in G': $\quad [A, a_0] \to \langle x_1 \dots x_m \rangle([A_1, a_1], \dots, [A_m, a_m])$

The canonical weighted deduction system

- wRTG-LM $((G, \mathcal{L}), \mathbb{K}, \mathrm{wt})$
- $a \in \mathcal{L}$

$$\xrightarrow{\text{cwds}} \text{wRTG-LM } ((G', \mathcal{CFG}^{\emptyset}), \mathbb{K}, \mathrm{wt}')$$

Parsing as deduction (Shieber, Schabes, and Pereira 1995)

$$\frac{[A_1, a_1] \ \dots \ [A_m, a_m]}{[A, a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule in } G \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \phi(\sigma)(a_1, \dots, a_m) \end{cases}$$

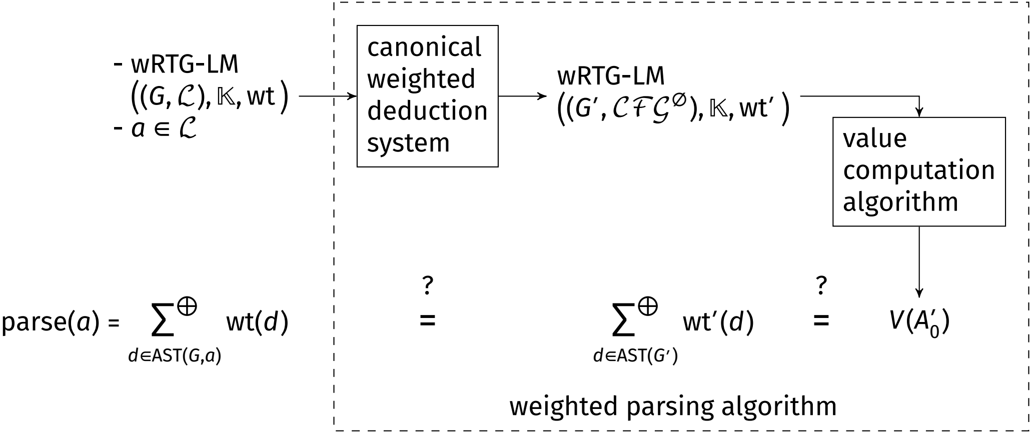
Rule in G': $\quad [A, a_0] \to \langle x_1 \dots x_m \rangle([A_1, a_1], \dots, [A_m, a_m])$

Weight preserving

1. Bijection $\psi\colon \mathrm{AST}(G, a) \to \mathrm{AST}(G')$
2. $\mathrm{wt}(d) = \mathrm{wt}'(\psi(d))$ for every $d \in \mathrm{AST}(G, a)$

The M-monoid parsing algorithm

Two-phase pipeline (Goodman 1999; Nederhof 2003)



The value computation algorithm

- Infinite summation vs. termination

The value computation algorithm

- Infinite summation vs. termination
- Single-source shortest distance framework by Mohri (2002)

The value computation algorithm

- Infinite summation vs. termination
- Single-source shortest distance framework by Mohri (2002)
  - weighted directed graph $G = (V, E, \mathrm{wt})$ over commutative semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$

The value computation algorithm

- Infinite summation vs. termination
- Single-source shortest distance framework by Mohri (2002)
  - weighted directed graph $G = (V, E, \mathrm{wt})$ over commutative semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
  - $P(v)$ = set of paths from the source to $v \in V$

The value computation algorithm

- Infinite summation vs. termination
- Single-source shortest distance framework by Mohri (2002)
  - weighted directed graph $G = (V, E, \mathrm{wt})$ over commutative semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
  - $P(v)$ = set of paths from the source to $v \in V$
  - $P_n(v)$ = set of paths with at most n occurrences of a cycle

The value computation algorithm

- Infinite summation vs. termination
- Single-source shortest distance framework by Mohri (2002)
  - weighted directed graph $G = (V, E, \mathrm{wt})$ over commutative semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
  - $P(v)$ = set of paths from the source to $v \in V$
  - $P_n(v)$ = set of paths with at most n occurrences of a cycle

The value computation algorithm

- Infinite summation vs. termination
- Single-source shortest distance framework by Mohri (2002)
  - weighted directed graph $G = (V, E, \mathrm{wt})$ over commutative semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
  - $P(v)$ = set of paths from the source to $v \in V$
  - $P_n(v)$ = set of paths with at most n occurrences of a cycle

$$\mathbb{K} \text{ is } c\text{-closed for } G: \qquad \exists c \in \mathbb{N}: \sum_{p \in P(v)}^{\oplus} \mathrm{wt}(p) = \bigoplus_{p \in P_c(v)} \mathrm{wt}(p)$$

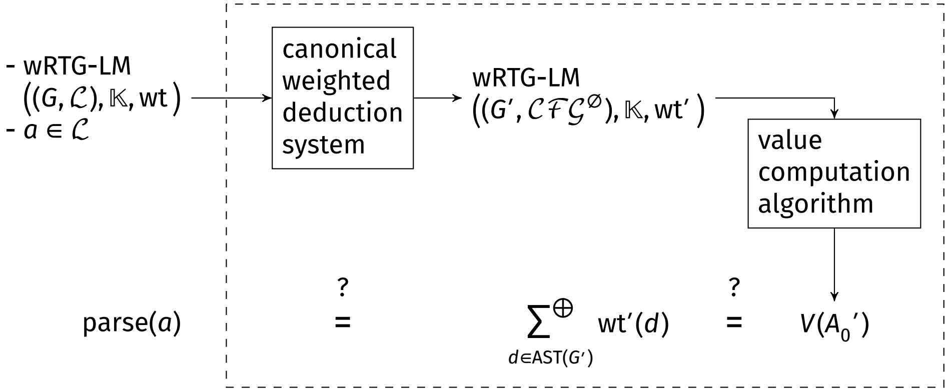
The value computation algorithm

- Infinite summation vs. termination
- Single-source shortest distance framework by Mohri (2002)
  - weighted directed graph $G = (V, E, \mathrm{wt})$ over commutative semiring $(\mathbb{K}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
  - $P(v)$ = set of paths from the source to $v \in V$
  - $P_n(v)$ = set of paths with at most n occurrences of a cycle

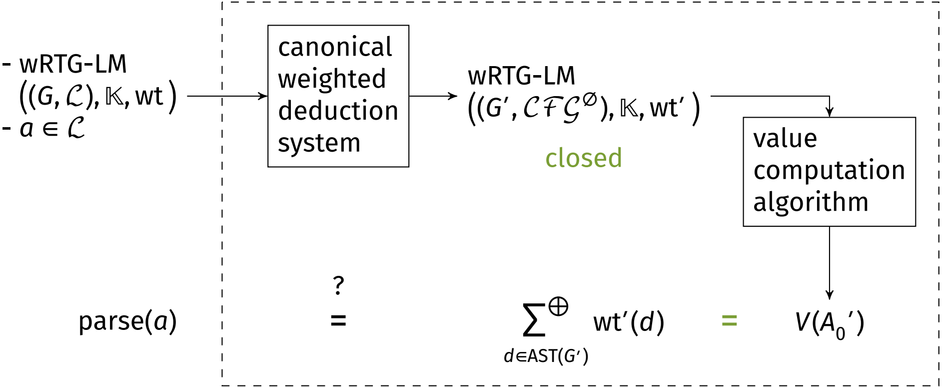
$$\mathbb{K} \text{ is } c\text{-closed for } G: \qquad \exists c \in \mathbb{N}\colon \sum^{\oplus}_{p \in P(v)} \mathrm{wt}(p) = \bigoplus_{p \in P_c(v)} \mathrm{wt}(p)$$

- Generalization to our framework
  - introduce branching (strings ⇝ trees)
  - requires distributivity

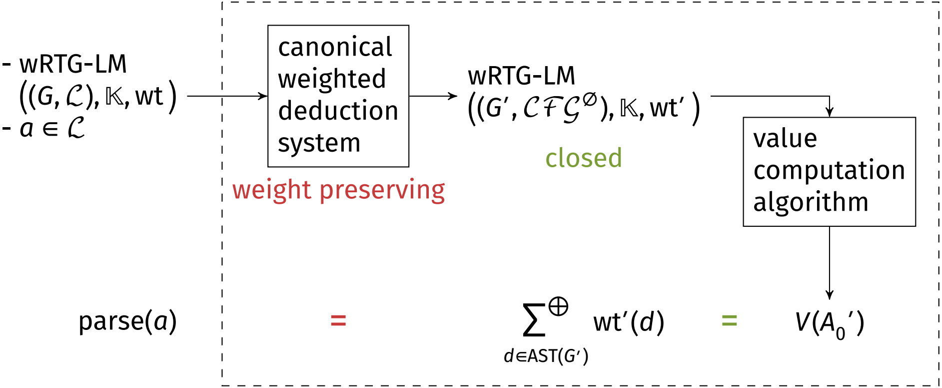
Application scenarios



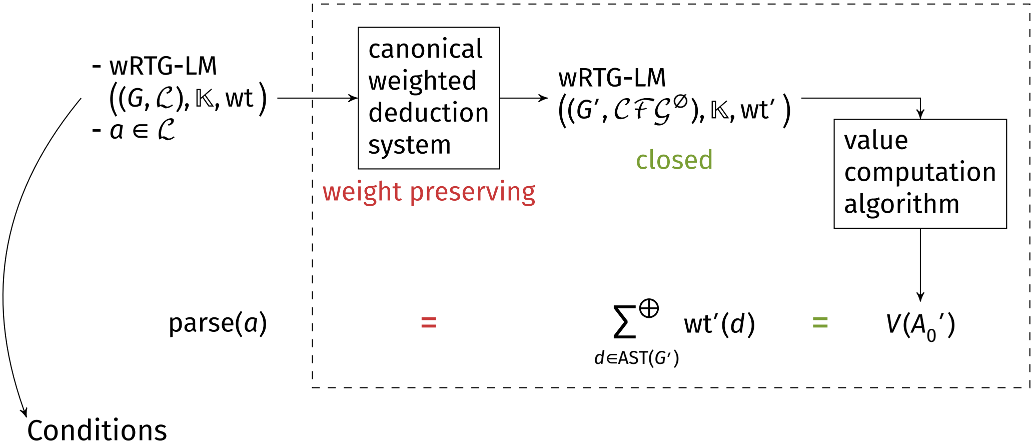
Application scenarios



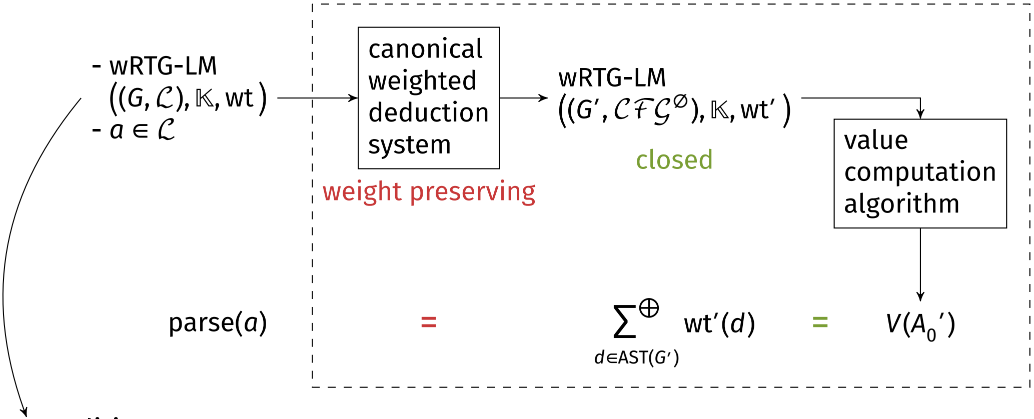
Application scenarios



Application scenarios



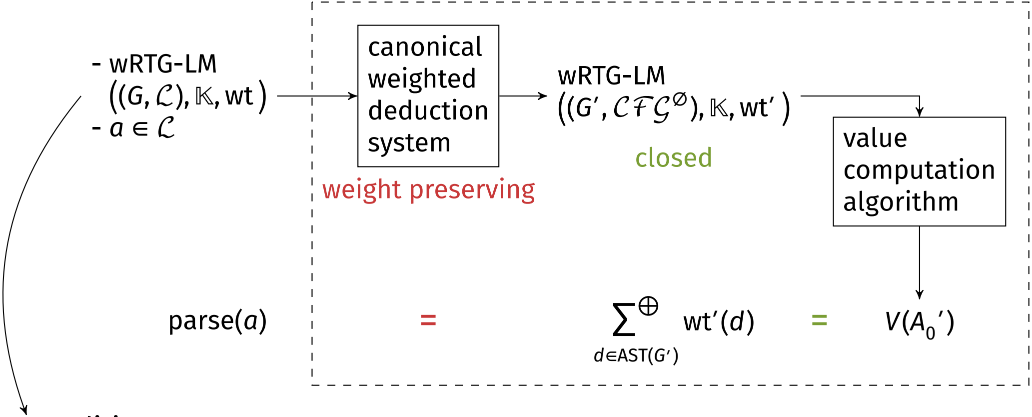
Application scenarios



Conditions

- Sufficient: $((G, \mathcal{L}), \mathbb{K}, \mathrm{wt})$ is closed or nonlooping

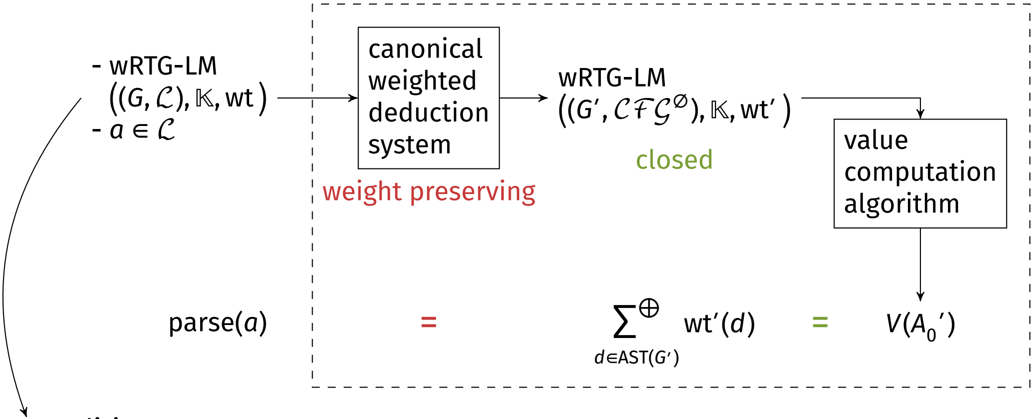
Application scenarios



Conditions

- Sufficient: $((G, \mathcal{L}), \mathbb{K}, \mathrm{wt})$ is closed or nonlooping
- $\mathcal{L}$ is finitely decomposable

Application scenarios



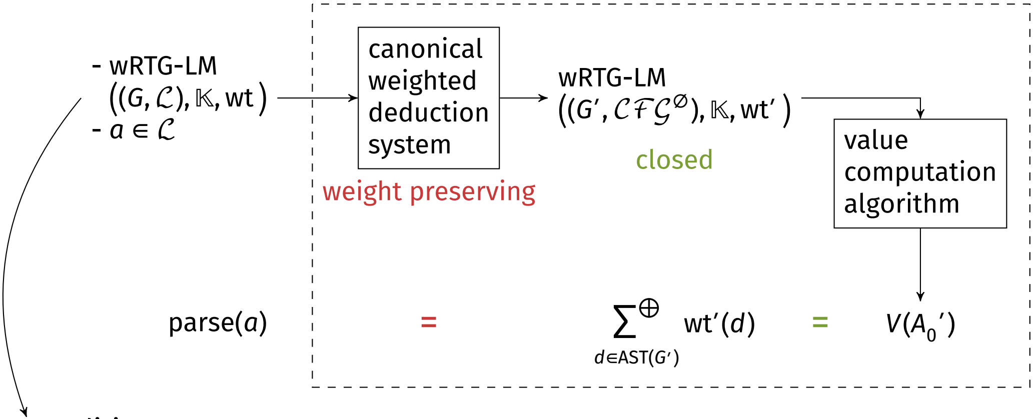
Conditions

- Sufficient: $((G, \mathcal{L}), \mathbb{K}, \mathrm{wt})$ is closed or nonlooping

  e.g., acyclic RTGs, superior M-monoids, algebraic dynamic programming (Giegerich, Meyer, and Steffen 2004)

- $\mathcal{L}$ is finitely decomposable

Application scenarios



Conditions

- Sufficient: $((G, \mathcal{L}), \mathbb{K}, \mathrm{wt})$ is closed or nonlooping

  e.g., acyclic RTGs, superior M-monoids, algebraic dynamic programming (Giegerich, Meyer, and Steffen 2004)

- $\mathcal{L}$ is finitely decomposable

  e.g., CFG, LCFRS, TAG

References I

R. Giegerich, C. Meyer, and P. Steffen (2004). “A discipline of dynamic programming over sequence data”. *Science of Computer Programming.*

J. Goodman (1999). “Semiring Parsing”. *Computational Linguistics, 4.*

M. Mohri (2002). “Semiring frameworks and algorithms for shortest-distance problems”. *Journal of Automata, Languages and Combinatorics.*

M.-J. Nederhof (2003). “Squibs and Discussions: Weighted deductive parsing and Knuth’s algorithm”. *Computational Linguistics.*

S. Shieber, Y. Schabes, and F. Pereira (1995). “Principles and implementation of deductive parsing”. *The Journal of Logic Programming.*

Canonical weighted deduction system

wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$ and $a \in \mathcal{L} \rightsquigarrow$ wRTG-LM $\big((G', \mathcal{CFG}^{\varnothing}), \mathbb{K}, \mathrm{wt}'\big)$

$$\frac{[A_1, \quad a_1] \ \dots \ [A_m, \quad a_m]}{[A, \quad a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule} \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \sigma(a_1, \dots, a_m) \\ \end{cases}$$

Canonical weighted deduction system

wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$ and $a \in \mathcal{L} \rightsquigarrow$ wRTG-LM $\big((G', \mathcal{CFG}^{\varnothing}), \mathbb{K}, \mathrm{wt}'\big)$

$$\frac{[A_1, \sigma_1, a_1] \; \dots \; [A_m, \sigma_m, a_m]}{[A, \sigma, a_0]} \quad \begin{cases} A \to \sigma(A_1, \dots, A_m) \text{ is a rule} \\ a_0, a_1, \dots, a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \sigma(a_1, \dots, a_m) \\ A_1 \to \sigma_1(\dots), \dots, A_m \to \sigma_m(\dots) \text{ are rules} \end{cases}$$

Canonical weighted deduction system

$$\text{wRTG-LM}\big((G,\mathcal{L}),\mathbb{K},\mathrm{wt}\big) \text{ and } a \in \mathcal{L} \rightsquigarrow \text{wRTG-LM}\big((G',\mathcal{CFG}^{\varnothing}),\mathbb{K},\mathrm{wt}'\big)$$

$$\frac{[A_1,\sigma_1,a_1]\ \dots\ [A_m,\sigma_m,a_m]}{[A,\sigma,a_0]} \quad \begin{cases} A \to \sigma(A_1,\dots,A_m) \text{ is a rule} \\ a_0,a_1,\dots,a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L} \\ a_0 = \sigma(a_1,\dots,a_m) \\ A_1 \to \sigma_1(\dots),\dots,A_m \to \sigma_m(\dots) \text{ are rules} \end{cases}$$

$$\frac{[A_0,\sigma,a]}{[A_0,a]} \quad \{A_0 \to \sigma(\dots) \text{ is a rule}$$

Canonical weighted deduction system

wRTG-LM $\big((G,\mathcal{L}),\mathbb{K},\mathrm{wt}\big)$ and $a \in \mathcal{L} \rightsquigarrow$ wRTG-LM $\big((G',\mathcal{CFG}^{\varnothing}),\mathbb{K},\mathrm{wt}'\big)$

$$\frac{[A_1,\sigma_1,a_1]\ \dots\ [A_m,\sigma_m,a_m]}{[A,\sigma,a_0]}\quad \begin{cases} A \to \sigma(A_1,\dots,A_m) \text{ is a rule}\\ a_0,a_1,\dots,a_m \text{ are part of a decomposition of } a \text{ in } \mathcal{L}\\ a_0 = \sigma(a_1,\dots,a_m)\\ A_1 \to \sigma_1(\dots),\dots,A_m \to \sigma_m(\dots) \text{ are rules}\end{cases}$$

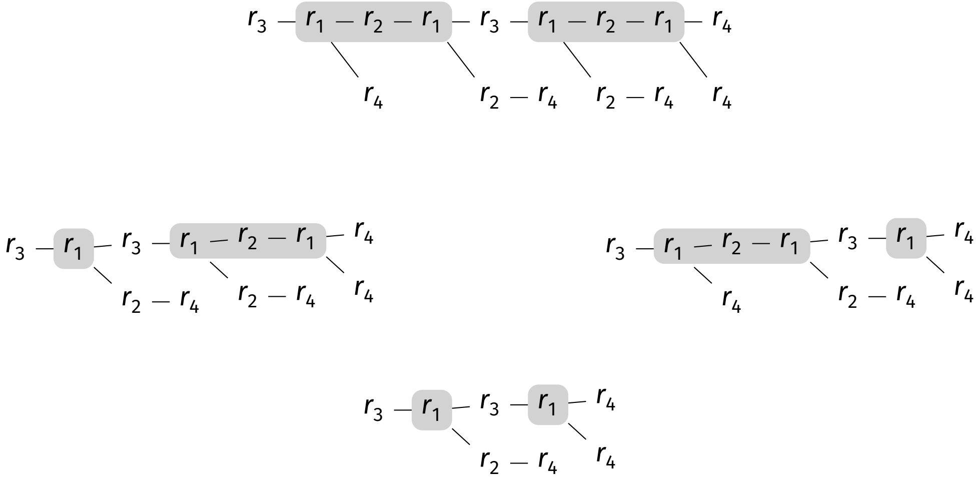
$$\frac{[A_0,\sigma,a]}{[A_0,a]}\quad \{A_0 \to \sigma(\dots) \text{ is a rule}$$

Weight preserving

1. Bijection $\psi\colon \mathrm{AST}(G,a) \to \mathrm{AST}(G')$
2. $\mathrm{wt}(d) = \mathrm{wt}'(\psi(d))$ for every $d \in \mathrm{AST}(G,a)$

Closed wRTG-LMs

cutout(d, ρ)



Closed wRTG-LMs

Definition

Let $c \in \mathbb{N}$. A wRTG-LM $\mathcal{G} = \big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$ is *c-closed* if $\mathbb{K}$ is distributive and d-complete, and for each $d \in \mathrm{T}_R$ and cyclic string $\rho \in R^*$ the following holds: if there is a (c, ρ)-cyclic path in d, then

$$\mathrm{wt}(d)_{\mathbb{K}} \oplus \bigoplus_{d \in \mathrm{cutout}(d,\rho)} \mathrm{wt}(d)_{\mathbb{K}} = \bigoplus_{d \in \mathrm{cutout}(d,\rho)} \mathrm{wt}(d)_{\mathbb{K}} \ .$$

$\mathrm{AST}(G)^{(c)}$: each cycle at most c times

↓ closed, distributive, d-complete

Theorem

For every $c \in \mathbb{N}$ and c-closed wRTG-LM $\big((G, \mathcal{L}), \mathbb{K}, \mathrm{wt}\big)$ the following holds:

$$\sum^{\oplus}_{d \in \mathrm{AST}(G')} \mathrm{wt}(d)_{\mathbb{K}} = \bigoplus_{d \in \mathrm{AST}(G)^{(c)}} \mathrm{wt}(d)_{\mathbb{K}} \ .$$

The value computation algorithm

Input: a wRTG-LM $\big((G', \mathcal{CFG}^{\varnothing}), (\mathbb{K}, \oplus, \mathbb{0}, \Omega), \mathrm{wt}'\big)$ with $G' = (N', \Sigma', A'_0, R')$
Variables: $V\colon N' \to \mathbb{K}$, $V_{\text{new}} \in \mathbb{K}$, $\mathit{changed} \in \mathbb{B}$
Output: $V(A'_0)$

for each $A \in N'$ **do**
  $V(A) \leftarrow \mathbb{0}$
repeat
  $\mathit{changed} \leftarrow$ false
  for each $A \in N'$ **do**
    $V_{\text{new}} \leftarrow \mathbb{0}$
    for each $r = (A \to \langle x_1 \dots x_m \rangle(A_1, \dots, A_m))$ in R' **do**
      $V_{\text{new}} \leftarrow V_{\text{new}} \oplus \mathrm{wt}'(r)\big(V(A_1), \dots, V(A_m)\big)$
    if $V(A) \neq V_{\text{new}}$ **then**
      $\mathit{changed} \leftarrow$ true
    $V(A) \leftarrow V_{\text{new}}$
until $\mathit{changed}$ = false