Open access • Book Chapter • DOI:10.1007/978-3-540-71322-7_9

Grammar analysis and parsing by abstract interpretation — Source link

Patrick Cousot, Radhia Cousot

Institutions: École Normale Supérieure, École Polytechnique

Published on: 01 Jan 2007

Topics: Mildly context-sensitive grammar formalism, S-attributed grammar, Formal semantics (linguistics), Operational semantics and Context-free language

Grammar Analysis and Parsing by Abstract Interpretation

Patrick Cousot[1] and Radhia Cousot[2]

[1] École Normale Supérieure, 45 rue d'Ulm, 75230 Paris cedex 05 (France)
Patrick.Cousot@ens.fr www.di.ens.fr/~cousot

[2] CNRS & École polytechnique, 91128 Palaiseau cedex (France)
Radhia.Cousot@polytechnique.fr
www.enseignement.polytechnique.fr/profs/informatique/Radhia.Cousot/

Abstract. We study abstract interpretations of a fixpoint protoderivation semantics defining the maximal derivations of a transitional semantics of context-free grammars akin to pushdown automata. The result is a hierarchy of bottom-up or top-down semantics refining the classical equational and derivational language semantics and including Knuth grammar problem, classical grammar flow analysis algorithms, and parsing algorithms.

1 Introduction

Grammar flow problems consist in computing a function of the [proto]language generated by the grammar for each nonterminal. This includes Knuth's grammar problem [1,2], grammar decision problems such as emptiness and finiteness [3], and classical compilation algorithms such as First and Follow [4]. For the later case, Ulrich Möncke and Reinhard Wilhelm introduced *grammar flow analysis* to solve computation problems over context-free grammars [5,6,7], [8, Sect. 8.2.4]. The idea is to provide two fixpoint algorithm schemata, one for bottom-up grammar flow analysis and one for top-down grammar flow analysis which can be instantiated with different parameters to get classical iterative algorithms such as First and Follow.

More generally, we show that grammar flow algorithms are abstract interpretations [9] of a hierarchy of bottom-up or top-down grammar semantics refining the classical (proto-)language semantics.

Then, we apply this comprehensive abstract-interpretation-based approach to the systematic derivation of parsing algorithms.

2 Languages and Context-free Grammars

A *sentence* $\sigma \in \mathcal{A}^\star$ *over the alphabet* $\mathcal{A}$ of length $|\sigma| \triangleq n \geqslant 0$ is a possibly empty finite sequence $\sigma_1\sigma_2\ldots\sigma_n$ of letters $\sigma_1, \sigma_2, \ldots, \sigma_n \in \mathcal{A}$. For $n = 0$, the empty sentence is denoted ϵ of length $|\epsilon| = 0$. A *language* Σ *over the alphabet* $\mathcal{A}$

is a set of sentences $\Sigma \in \wp(\mathcal{A}^\star)$. We represent concatenation by juxtaposition. It is extended to languages as $\Sigma\Sigma' \triangleq \{\sigma\sigma' \mid \sigma \in \Sigma \wedge \sigma' \in \Sigma'\}$. For brevity, σ denotes the language $\{\sigma\}$ so that we can write $\Sigma\sigma\Sigma'$ for $\Sigma\{\sigma\}\Sigma'$. The *junction* of languages is $\Sigma \fatsemi \Sigma' \triangleq \{\sigma_1\sigma_2 \ldots \sigma_m\sigma'_2 \ldots \sigma'_n \mid \sigma_1\sigma_2 \ldots \sigma_m \in \Sigma \wedge \sigma'_1\sigma'_2 \ldots \sigma'_n \in \Sigma' \wedge \sigma_m = \sigma'_1\}$. Given a set $\mathscr{P} \triangleq \{[_i \mid i \in \Delta\} \cup \{]_i \mid i \in \Delta\}$ of matching parentheses and an alphabet $\mathcal{A}$, the *Dyck language* $\mathbb{D}_{\mathscr{P},\mathcal{A}} \subseteq (\mathscr{P} \cup \mathcal{A})^\star$ over $\mathscr{P}$ and $\mathcal{A}$ is the set of well-parenthesized sentences over $\mathscr{P} \cup \mathcal{A}$. It is *pure* if $\mathcal{A} = \varnothing$. The *parenthesized language* over $\mathscr{P}$ and $\mathcal{A}$ is $\mathbb{P}_{\mathscr{P},\mathcal{A}} \triangleq \{[_i \sigma]_i \mid i \in \Delta \wedge \sigma \in \mathbb{D}_{\mathscr{P},\mathcal{A}} \setminus \{\epsilon\}\}$.

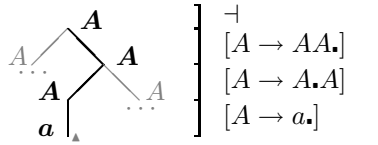
A context-free grammar [10,11] is a quadruple $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ where $\mathscr{T}$ is the alphabet of *terminals*, $\mathscr{N}$ such that $\mathscr{T} \cap \mathscr{N} = \varnothing$ is the alphabet of *nonterminals*, $\overline{S} \in \mathscr{N}$ is the *start symbol* (or *axiom*) and $\mathscr{R} \in \wp(\mathscr{N} \times \mathscr{V}^\star)$ is the finite set of *rules* written $A \to \sigma$ where the *lefthand side* $A \in \mathscr{N}$ is a nonterminal and the *righthand side* $\sigma \in \mathscr{V}^\star$ is a possibly empty sentence over the *vocabulary* $\mathscr{V} \triangleq \mathscr{T} \cup \mathscr{N}$. By convention, $\epsilon \notin \mathscr{V}$.

3 Transitional Semantics of Context-free Grammars

Pushdown automata (PDA) and context-free grammars are equivalent [8, Sect. 8.2]. Inspired by PDA, we define the transitional semantics of grammars by labelled transition systems where states are stacks, labels encode the structure of sentences and transitions are small steps in the recursive derivation of sentences.

Stacks. Given a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$, we let *stacks* $\varpi \in \mathcal{S} \triangleq (\mathscr{R}^\bullet \cup \mathscr{M})^\star$ be sentences over *rule states* $\mathscr{R}^\bullet \triangleq \{[A \to \sigma\mathbf{.}\sigma'] \mid A \to \sigma\sigma' \in \mathscr{R}\}$ specifying the state of the derivation (σ' is still to be derived) and markers $\mathscr{M} = \{\vdash, \dashv\}$ where $\vdash$ (resp. $\dashv$) marks the beginning (resp. the end) of a sentence. The *height* of a stack ϖ is its length $|\varpi|$.

Example 1 A stack ϖ for the grammar $A \to AA$, $A \to a$ is $\dashv[A \to AA\mathbf{.}][A \to A\mathbf{.}A][A \to a\mathbf{.}]$. It records the ancestors in an infix traversal of a parse tree, as shown opposite.



□

Labels. We let $\mathscr{P} \triangleq \mathscr{O} \cup \mathscr{C}$ be the set of *parentheses* where $\mathscr{O} \triangleq \{(\!|A| \mid A \in \mathscr{N}\}$ is the set of *opening parentheses* while $\mathscr{C} \triangleq \{A|\!) \mid A \in \mathscr{N}\}$ is the set of *closing parentheses*. We let *labels* $\ell \in \mathscr{L}$ be parentheses or terminals so that $\mathscr{L} \triangleq \mathscr{P} \cup \mathscr{T}$. A pair of parentheses $(\!|A \ldots A|\!)$ delimits the structure of a sentence deriving from nonterminal $A \in \mathscr{N}$ while terminals describe elements of the sentence.

Labelled Transition System. Given a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$, we define a *labelled transition system* $\mathsf{S}^t[\![\mathcal{G}]\!] \triangleq \langle \mathcal{S}, \mathscr{L}, \longrightarrow, \vdash \rangle$ where the initial state is $\vdash$ and the labelled transition relation $\xrightarrow{\ell}$, $\ell \in \mathscr{L}$ is

$$\vdash \xrightarrow{(\!|A} \dashv[A \to \centerdot\sigma], \qquad A \to \sigma \in \mathscr{R} \tag{1}$$

$$\varpi[A \to \sigma\centerdot a\sigma'] \xrightarrow{a} \varpi[A \to \sigma a\centerdot\sigma'], \qquad A \to \sigma a\sigma' \in \mathscr{R} \tag{2}$$

$$\varpi[A \to \sigma\centerdot B\sigma'] \xrightarrow{(\!|B} \varpi[A \to \sigma B\centerdot\sigma'][B \to \centerdot\varsigma], \quad A \to \sigma B\sigma' \in \mathscr{R} \wedge B \to \varsigma \in \mathscr{R} \tag{3}$$

$$\varpi[A \to \sigma\centerdot] \xrightarrow{A|)} \varpi, \qquad A \to \sigma \in \mathscr{R}\ . \tag{4}$$

4 Maximal Derivations

The *maximal derivation semantics* of a grammar is the set of all possible maximal derivations for this grammar where a *maximal derivation* is a finite labelled trace of maximal length generated by the transitional semantics.

Example 2 The maximal derivation for the sentence a of the grammar $\langle\{a\}, \{A\}, A, \{A \to AA, A \to a\}\rangle$ is $\vdash \xrightarrow{(\!|A} \dashv [A \to \centerdot a] \xrightarrow{a} \dashv [A \to a\centerdot] \xrightarrow{A|)} \dashv$ while for the sentence aa it is $\vdash \xrightarrow{(\!|A} \dashv [A \to \centerdot AA] \xrightarrow{(\!|A} \dashv [A \to A\centerdot A][A \to \centerdot a] \xrightarrow{a} \dashv [A \to A\centerdot A][A \to a\centerdot] \xrightarrow{A|)} \dashv [A \to A\centerdot A] \xrightarrow{(\!|A} \dashv [A \to AA\centerdot][A \to \centerdot a] \xrightarrow{a} \dashv [A \to AA\centerdot][A \to a\centerdot] \xrightarrow{A|)} \dashv [A \to AA\centerdot] \xrightarrow{A|)} \dashv$. □

Traces. Formally a *trace* $\theta \in \Theta^n$ of length $|\theta| = n + 1$, $n \geqslant 0$, has the form $\theta = \varpi_0 \xrightarrow{\ell_0} \varpi_1 \ldots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n$ whence it is a pair $\theta = \langle \underline{\theta}, \overline{\theta}\rangle$ where $\underline{\theta} \in [0, n] \mapsto \mathcal{S}$ is a nonempty finite sequence of stacks $\underline{\theta}_i = \varpi_n$, $i = 0, \ldots, n$ and $\overline{\theta} \in [0, n-1] \mapsto \mathscr{L}$ is a finite sequence of labels $\overline{\theta}_j = \ell_j$, $j = 0, \ldots, n-1$. Traces $\theta \in \Theta$ are nonempty, finite, of any length so $\Theta \triangleq \bigcup_{n \geqslant 0} \Theta^n$.

Again concatenation is denoted by juxtaposition and extended to sets. We respectively identify a single state ϖ and a transition $\varpi \xrightarrow{\ell} \varpi'$ with the corresponding traces containing only the single state ϖ and the the transition $\varpi \xrightarrow{\ell} \varpi'$. By abuse of notation, a trace $\varpi_0 \xrightarrow{\ell_0} \varpi_1 \ldots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n$ is also understood as the concatenation of ϖ_0, $\xrightarrow{\ell_0}$, ϖ_1, …, ϖ_{n-1}, $\xrightarrow{\ell_{n-1}}$, ϖ_n which, informally, *matches the trace pattern* $\varsigma_0\varpi_1 \ldots \varsigma_{n-1}\varpi_n\varsigma_n$ by letting $\varsigma_0 = \varpi_0 \xrightarrow{\ell_0}$, …, $\varsigma_{n-1} = \varpi_{n-1} \xrightarrow{\ell_{n-1}}$ and $\varsigma_n = \epsilon$. We also need the *junction* of sets of traces, as follows

$$T \fatsemi T' \triangleq \{\theta \xrightarrow{\ell} \varpi \xrightarrow{\ell'} \theta' \mid \theta \xrightarrow{\ell} \varpi \in T \wedge \varpi' \xrightarrow{\ell'} \theta' \in T' \wedge \varpi = \varpi'\}\ .$$

The *selection* of the traces in T for nonterminal B is denoted $T.B$ defined as

$$T.B \triangleq \{\varpi \xrightarrow{(\!|B} \theta \mid \varpi \xrightarrow{(\!|B} \theta \in T\}\ .$$

For the recursive *incorporation* of a derivation $\vdash \xrightarrow{\ell_0} \dashv\varpi_1 \ldots \dashv\varpi_{n-1} \xrightarrow{\ell_{n-1}} \dashv$ into another one, we need the operation

$$\langle\varpi, \varpi'\rangle \uparrow \vdash \xrightarrow{\ell_0} \dashv\varpi_1 \ldots \dashv\varpi_{n-1} \xrightarrow{\ell_{n-1}} \dashv \triangleq \varpi \xrightarrow{\ell_0} \varpi'\varpi_1 \ldots \varpi'\varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi'$$

$$\langle \varpi,\, \varpi' \rangle \uparrow T \triangleq \{ \langle \varpi,\, \varpi' \rangle \uparrow \tau \mid \tau \in T \} \; .$$

Example 3 We have $\langle \dashv[A \to \bullet AA],\, \dashv[A \to A\bullet A] \rangle \uparrow \vdash \xrightarrow{(\!|A} \dashv[A \to \bullet a] \xrightarrow{a} \dashv[A \to a\bullet] \xrightarrow{A|)} \dashv = \dashv[A \to \bullet AA] \xrightarrow{(\!|A} \dashv[A \to A\bullet A][A \to \bullet a] \xrightarrow{a} \dashv[A \to A\bullet A][A \to a\bullet] \xrightarrow{A|)} \dashv[A \to A\bullet A]$ which we can recognize as the replacement of the first A deriving into a in the derivation for the sentence aa in **Ex. 2**. □

A *derivation* of grammar $\mathcal{G}$ is a trace $\varpi_0 \xrightarrow{\ell_0} \varpi_1 \ldots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n$, $n \geqslant 0$ generated by the transition system $\mathsf{S}^t[\![\mathcal{G}]\!]$ that is $\forall i \in [0, n-1] : \varpi_i \xrightarrow{\ell_i} \varpi_{i+1}$. A *prefix derivation* of grammar $\mathcal{G}$ is a derivation of grammar $\mathcal{G}$ starting with an initial state $\varpi_0 = \vdash$. A *suffix derivation* of grammar $\mathcal{G}$ is derivation of grammar $\mathcal{G}$ ending with an final state $\forall \varpi \in \mathcal{S} : \forall \ell \in \mathscr{L} : \neg(\varpi_n \xrightarrow{\ell} \varpi)$, so that $\varpi_n = \dashv$ by def. (1–4) of $\longrightarrow$. A *maximal derivation* of grammar $\mathcal{G}$ is both a prefix and a suffix derivation of the grammar $\mathcal{G}$.

Derivations are well-parenthesized so that the grammatical structure of sentences can be described by trees. Let us define the *parenthesis abstraction* α^p for a stack ϖ by $\alpha^p(\varpi\varpi') \triangleq \alpha^p(\varpi')\alpha^p(\varpi)$, $\alpha^p(\vdash) = \alpha^p(\dashv) = \epsilon$ and $\alpha^p([A \to \sigma\bullet\sigma']) \triangleq A|)$, for a label, $\alpha^p(a) \triangleq \epsilon$ for all $a \in \mathscr{T}$, $\alpha^p((\!|A) \triangleq (\!|A$ and $\alpha^p(A|)) \triangleq A|)$, and for a trace $\alpha^p(\varpi_0 \xrightarrow{\ell_0} \varpi_1 \xrightarrow{\ell_1} \ldots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n) \triangleq \alpha^p(\ell_0)\alpha^p(\ell_1)\ldots\alpha^p(\ell_{n-1})\alpha^p(\varpi_n)$.

Lemma 4 *For any prefix derivation θ of a grammar $\mathcal{G}$, $\alpha^p(\theta) \in \mathbb{D}_{\mathscr{P},\varnothing}$ is a pure Dyck language. A maximal derivation $\theta = \vdash \xrightarrow{\ell_0} \varpi_1 \xrightarrow{\ell_1} \ldots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \dashv$ of $\mathcal{G}$ is well-parenthesized in that $\alpha^p(\theta) = \alpha^p(\ell_0)\alpha^p(\ell_1)\ldots\alpha^p(\ell_{n-1}) \in \mathbb{D}_{\mathscr{P},\varnothing}$ is a pure Dyck language.* □

5 Prefix Derivation Semantics

The *prefix derivation semantics* $\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!]$ of a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ is the set of all prefix derivations for the labelled transition system $\langle \mathcal{S}, \mathscr{L}, \longrightarrow, \vdash \rangle$, that is

$$\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!] \triangleq \{ \varpi_0 \xrightarrow{\ell_0} \varpi_1 \ldots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n \mid n > 0 \wedge \varpi_0 = \vdash \wedge \forall i \in [0, n-1] : \varpi_i \xrightarrow{\ell_i} \varpi_{i+1} \} \; .$$

Lemma 5 *If the prefix derivation semantics $\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!]$ of a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ contains a prefix derivation $\theta_1\varpi\theta_2$ then*

- *either $\varpi = \vdash$ if and only if $\theta_1 = \epsilon$*
- *or the stack ϖ has the form $\varpi = \dashv[A_1 \to \eta_1 A_2 \bullet \eta_1'][A_2 \to \eta_2 A_3 \bullet \eta_2'] \ldots [A_n \to \eta_n \bullet \eta_n']$ where $A_i \to \eta_i A_{i+1} \eta_i' \in \mathscr{R}$ and $A_n \to \eta_i \eta_n' \in \mathscr{R}$ are grammar rules and $\theta_1 = \vdash \xrightarrow{(\!|A_1} \theta_1'$.*
- *Moreover if $\theta_1\varpi\theta_2 \in \mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!].A$ then necessarily $A_1 = A$.* □

It has been shown in the more general context of [12, **Th. 11**] that we have the following fixpoint characterization of the prefix derivation semantics

Theorem 6

$$\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!] = \mathit{lfp}^{\subseteq}\, \mathsf{F}^{\vec{\partial}}[\![\mathcal{G}]\!] = \mathit{gfp}^{\subseteq}\, \mathsf{F}^{\vec{\partial}}[\![\mathcal{G}]\!]$$

where $\mathsf{F}^{\vec{\partial}}[\![\mathcal{G}]\!] \in \wp(\Theta) \mapsto \wp(\Theta)$ *is a complete* $\cup$ *and* $\cap$ *morphism defined as*

$$\mathsf{F}^{\vec{\partial}}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, X \bullet \{\vdash\} \cup X ; \longrightarrow \ .$$

□

6 Transitional Maximal Derivation Semantics

The *maximal derivation semantics* $\mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!] \in \wp(\Theta)$ of a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ is the set of maximal derivations for the labelled transition system $\mathsf{S}^{t}[\![\mathcal{G}]\!] \triangleq \langle \mathcal{S}, \mathscr{L}, \longrightarrow, \vdash \rangle$.

$$\begin{aligned}\mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!] \triangleq \{ & \varpi_0 \xrightarrow{\ell_0} \varpi_1 \dots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n \mid n > 0 \wedge \varpi_0 = \vdash \wedge \\ & \forall i \in [0, n-1] : \varpi_i \xrightarrow{\ell_i} \varpi_{i+1} \wedge \forall \varpi \in \mathcal{S} : \forall \ell \in \mathscr{L} : \neg(\varpi_n \xrightarrow{\ell} \varpi)\} \ .\end{aligned} \tag{5}$$

Lemma 7 *A maximal derivation of the transition system* $\mathsf{S}^{t}[\![\mathcal{G}]\!]$ *has the form* $\vdash \xrightarrow{\langle\!| A} \dashv[A \to \boldsymbol{.}\sigma] \xrightarrow{\ell_1} \dashv\varpi_2 \dots \dashv\varpi_{n-1} \xrightarrow{A|\!\rangle} \dashv$ *where* $\varpi_{n-1} \neq \epsilon$. □

7 Bottom-Up Fixpoint Maximal Derivation Semantics

The maximal derivation semantics (5) can be expressed in fixpoint form.

Example 8 For the grammar $\mathcal{G} = \langle \{a, b\}, \{A\}, A, \{A \to aA, A \to b\} \rangle$, we have $\mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!] = \mathbf{lfp}^{\subseteq} \vec{\hat{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!]$ where

$$\begin{aligned}\vec{\hat{\mathsf{F}}}^{\hat{d}}(T) \triangleq & \vdash \xrightarrow{\langle\!| A} \dashv[A \to \boldsymbol{.}b] \xrightarrow{b} \dashv[A \to b\boldsymbol{.}] \xrightarrow{A|\!\rangle} \dashv \cup \\ & \vdash \xrightarrow{\langle\!| A} (\dashv[A \to \boldsymbol{.}aA]) \xrightarrow{a} (\langle \dashv[A \to a\boldsymbol{.}A],\ \dashv[A \to aA\boldsymbol{.}] \rangle \uparrow T.A) \, ; \, (\dashv[A \to aA\boldsymbol{.}]) \xrightarrow{A|\!\rangle} \dashv.\end{aligned}$$

The first iterates of $\vec{\hat{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!]$ from $\vec{\hat{\mathsf{F}}}^{\hat{d}}{}_0 = \varnothing$ are

$$\begin{aligned}\vec{\hat{\mathsf{F}}}^{\hat{d}}{}_1 &= \{\vdash \xrightarrow{\langle\!| A} \dashv[A \to \boldsymbol{.}b] \xrightarrow{b} \dashv[A \to b\boldsymbol{.}] \xrightarrow{A|\!\rangle} \dashv\} \\ \vec{\hat{\mathsf{F}}}^{\hat{d}}{}_2 &= \{\vdash \xrightarrow{\langle\!| A} \dashv[A \to \boldsymbol{.}b] \xrightarrow{b} \dashv[A \to b\boldsymbol{.}] \xrightarrow{A|\!\rangle} \dashv, \\ &\quad \vdash \xrightarrow{\langle\!| A} \dashv[A \to \boldsymbol{.}aA] \xrightarrow{a} \dashv[A \to a\boldsymbol{.}A] \xrightarrow{\langle\!| A} \dashv[A \to aA\boldsymbol{.}][A \to \boldsymbol{.}b] \xrightarrow{b} \\ &\quad\ \dashv[A \to aA\boldsymbol{.}][A \to b\boldsymbol{.}] \xrightarrow{A|\!\rangle} \dashv[A \to aA\boldsymbol{.}] \xrightarrow{A|\!\rangle} \dashv\} \\ &\dots \quad \dots \\ \vec{\hat{\mathsf{F}}}^{\hat{d}}{}_\omega &= \mathbf{lfp}^{\subseteq} \vec{\hat{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!]\end{aligned}$$

□

More generally, let us define the set of traces bottom-up transformer $\hat{\vec{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!] \in \wp(\Theta) \mapsto \wp(\Theta)$ as

$$\hat{\vec{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, T \cdot \bigcup_{A \to \sigma \in \mathscr{R}} \vdash \xrightarrow{(\!|A} \hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \centerdot\sigma]T \xrightarrow{A|)} \dashv \tag{6}$$

where $\hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \sigma\centerdot\sigma'] \in \wp(\Theta) \mapsto \wp(\Theta)$ is defined as

$$\hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \sigma\centerdot a\sigma'] \triangleq \boldsymbol{\lambda}\, T \cdot (\dashv[A \to \sigma\centerdot a\sigma']) \xrightarrow{a} \hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \sigma a\centerdot\sigma']T \tag{7}$$

$$\hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \sigma\centerdot B\sigma'] \triangleq \boldsymbol{\lambda}\, T \cdot (\langle \dashv[A \to \sigma\centerdot B\sigma'],\ \dashv[A \to \sigma B\centerdot\sigma'] \rangle \uparrow T.B) \fatsemi \hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \sigma B\centerdot\sigma']T \tag{8}$$

$$\hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \sigma\centerdot] \triangleq \boldsymbol{\lambda}\, T \cdot (\dashv[A \to \sigma\centerdot]) \ . \tag{9}$$

Observe that $\hat{\vec{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!]$ is upper-continuous.

Lemma 9 *If all traces in $T \subseteq \Theta$ are derivations of the transition system $\mathsf{S}^t[\![\mathcal{G}]\!]$ then all traces in $\hat{\vec{\mathsf{F}}}^{\hat{d}}[A \to \sigma\centerdot\sigma']T$ are generated by the transition system $\mathsf{S}^t[\![\mathcal{G}]\!]$, start in state $(\dashv[A \to \sigma\centerdot\sigma'])$ and end in state $(\dashv[A \to \sigma\sigma'\centerdot])$. It follows that all traces in $\hat{\vec{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!]T$ are derivations of the transition system $\mathsf{S}^t[\![\mathcal{G}]\!]$.*

The derivation semantics of a grammar $\mathcal{G}$ can be expressed in fixpoint form as

Theorem 10 $\quad \mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!] = \mathit{lfp}^{\subseteq}\, \hat{\vec{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!]$. □

8 Protoderivations

Prototraces (formally defined below) are traces in construction containing nonterminal variables which are placeholders for unknown prototraces to be substituted for the nonterminal variables. Protoderivations are prototraces generated by the grammar, initially a nonterminal variable (such as the grammar axiom), obtained by top-down replacement of a nonterminal on the lefthand side of a grammar rule by the corresponding righthand side, until no nonterminal variable is left.

Example 11 A prototrace derivation for the grammar $\mathcal{G} = \langle \{a\}, \{A\}, A, \{A \to AA, A \to a\} \rangle$ is (the prototrace derivation relation is written $\mathrel{\mathbb{D}\!\!\Longrightarrow_{\mathcal{G}}}$)

$$\begin{aligned}
& \vdash \xrightarrow{\boxed{A}} \dashv \\
\mathrel{\mathbb{D}\!\!\Longrightarrow_{\mathcal{G}}}\ & \vdash \xrightarrow{(\!|A} \dashv[A \to \centerdot AA] \xrightarrow{\boxed{A}} \dashv[A \to A\centerdot A] \xrightarrow{\boxed{A}} \dashv[A \to AA\centerdot] \xrightarrow{A|)} \dashv \\
\mathrel{\mathbb{D}\!\!\Longrightarrow_{\mathcal{G}}}\ & \vdash \xrightarrow{(\!|A} \dashv[A \to \centerdot AA] \xrightarrow{\boxed{A}} \dashv[A \to A\centerdot A] \xrightarrow{(\!|A} \dashv[A \to AA\centerdot][A \to \centerdot a] \xrightarrow{a} \\
& \dashv[A \to AA\centerdot][A \to a\centerdot] \xrightarrow{A|)} \dashv[A \to AA\centerdot] \xrightarrow{A|)} \dashv \\
\mathrel{\mathbb{D}\!\!\Longrightarrow_{\mathcal{G}}}\ & \vdash \xrightarrow{(\!|A} \dashv[A \to \centerdot AA] \xrightarrow{(\!|A} \dashv[A \to A\centerdot A][A \to \centerdot a] \xrightarrow{a} \dashv[A \to A\centerdot A][A \to \\
& a\centerdot] \xrightarrow{A|)} \dashv[A \to A\centerdot A] \xrightarrow{(\!|A} \dashv[A \to AA\centerdot][A \to \centerdot a] \xrightarrow{a} \dashv[A \to AA\centerdot][A \to \\
& a\centerdot] \xrightarrow{A|)} \dashv[A \to AA\centerdot] \xrightarrow{A|)} \dashv \ .
\end{aligned}$$

□

Prototraces. The set of nonterminal variables is $\mathscr{N}^{\boxdot} \triangleq \{\boxed{A} \mid A \in \mathscr{N}\}$. A *prototrace* $\pi \in \mathit{\Pi}^n$ of length $|\pi| = n+1$, $n \geqslant 0$, has the form $\pi = \varpi_0 \xrightarrow{\kappa_0} \varpi_1 \ldots \varpi_{n-1} \xrightarrow{\kappa_{n-1}} \varpi_n$ whence is a pair $\pi = \langle \underline{\pi}, \overline{\pi} \rangle$ where $\underline{\pi} \in [0, n] \mapsto \mathcal{S}$ is a nonempty finite sequence of stacks $\underline{\pi}_i = \varpi_n$, $i = 0, \ldots, n$ and $\overline{\pi} \in [0, n-1] \mapsto (\mathscr{L} \cup \mathscr{N}^{\boxdot})$ is a finite sequence of labels or nonterminal variables $\overline{\pi}_j = \kappa_j$, $j = 0, \ldots, n-1$. Prototraces $\pi \in \mathit{\Pi}$ are nonempty, finite, of any length so $\mathit{\Pi} \triangleq \bigcup_{n \geqslant 0} \mathit{\Pi}^n$ and $\mathit{\Theta} \subseteq \mathit{\Pi}$.

Again prototrace pattern matching, prototrace concatenation, set of prototraces concatenation, the assimilation of a single state ϖ and a transition $\varpi \xrightarrow{\ell} \varpi'$ with the corresponding prototraces, the junction $\fatsemi$ of sets of prototraces, the selection $P.B$ of the prototraces in P for nonterminal B and the stack incorporation in a prototrace $\langle \varpi, \varpi' \rangle \uparrow \pi$ or a set T of prototraces $\langle \varpi, \varpi' \rangle \uparrow T$ are defined as for traces and sets of traces.

Prototrace Derivation. The *prototrace generated by a grammar rule* $A \to \sigma \in \mathscr{R}$ is $\check{\mathsf{R}}^{\check{D}}[A \to \sigma]$ where $\check{\mathsf{R}}^{\check{D}} \in \mathscr{R} \mapsto \mathit{\Pi}$ is

$$\begin{aligned}
\check{\mathsf{R}}^{\check{D}}[A \to \sigma] &\triangleq \vdash \xrightarrow{(\!|A} \check{\mathsf{F}}^{\check{D}}[A \to {\centerdot}\sigma] \xrightarrow{A|\!)} \dashv \\
\check{\mathsf{F}}^{\check{D}}[A \to \sigma{\centerdot}a\sigma'] &\triangleq \dashv[A \to \sigma{\centerdot}a\sigma'] \xrightarrow{a} \check{\mathsf{F}}^{\check{D}}[A \to \sigma a{\centerdot}\sigma'] \\
\check{\mathsf{F}}^{\check{D}}[A \to \sigma{\centerdot}B\sigma'] &\triangleq \dashv[A \to \sigma{\centerdot}B\sigma'] \xrightarrow{\boxed{B}} \check{\mathsf{F}}^{\check{D}}[A \to \sigma B{\centerdot}\sigma'] \\
\check{\mathsf{F}}^{\check{D}}[A \to \sigma{\centerdot}] &\triangleq \dashv[A \to \sigma{\centerdot}] \ .
\end{aligned} \tag{10}$$

The *prototrace derivation* relation $\boxdot\!\!\Longrightarrow_{\mathcal{G}} \in \wp(\mathit{\Pi} \times \mathit{\Pi})$ for a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ consists in replacing one or several nonterminal variables by the prototrace generated by a grammar rule for that nonterminal.

$$\begin{aligned}
&\pi \boxdot\!\!\Longrightarrow_{\mathcal{G}} \pi' \\
&\triangleq \exists n > 0, \varsigma_1, \ldots, \varsigma_{n+1}, \varpi_1, \ldots, \varpi_{n+1} \in \mathcal{S}, A_1, \ldots, A_n \in \mathscr{N}, \sigma_1, \ldots, \sigma_n \in \mathscr{V}^\star : \\
&\quad \pi = \varsigma_1 \varpi_1 \xrightarrow{\boxed{A_1}} \varpi_2 \varsigma_2 \ldots \varsigma_n \varpi_n \xrightarrow{\boxed{A_n}} \varpi_{n+1} \varsigma_{n+1} \wedge \forall i \in [1, n] : A_i \to \sigma_i \in \mathscr{R} \wedge \\
&\quad \pi' = \varsigma_1 \langle \varpi_1, \varpi_2 \rangle \uparrow \check{\mathsf{R}}^{\check{D}}[A_1 \to \sigma_1] \varsigma_2 \ldots \varsigma_n \langle \varpi_n, \varpi_{n+1} \rangle \uparrow \check{\mathsf{R}}^{\check{D}}[A_n \to \sigma_n] \varsigma_{n+1} \ .
\end{aligned} \tag{11}$$

9 Maximal Protoderivation Semantics

The *top-down maximal protoderivation semantics* $\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!] \in \mathscr{N} \mapsto \wp(\mathit{\Pi})$ of a context-free grammar $\mathcal{G}$ is

$$\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, A \bullet \{\pi \in \mathit{\Pi} \mid (\vdash \xrightarrow{\boxed{A}} \dashv) \boxdot\!\!\Longrightarrow_{\mathcal{G}}^{\star} \pi\} \ . \tag{12}$$

where $r^n, n \in \mathbb{N}$ are the powers of relation r, $r^{n\star} \triangleq \bigcup_{i<n} r^i$ (so that $r^{0\star} \triangleq \bigcup \varnothing = \varnothing$), r^+ (resp. $r^\star$) is the transitive closure (resp. reflexive transitive closure) of r.

10 Top-Down Fixpoint Maximal Protoderivation Semantics

The protoderivation semantics can be expressed in fixpoint form, as follows (where $\mathrm{post} \in \wp(\Sigma) \mapsto \wp(\Sigma)$ is $\mathrm{post}[r]X \triangleq \{s' \in \Sigma \mid \exists s \in X : \langle s, s' \rangle \in r\}$)

Theorem 12 $\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!] = \mathit{lfp}^{\dot{\subseteq}} \check{\mathsf{F}}^{\check{D}}[\![\mathcal{G}]\!]$ *where* $\dot{\subseteq}$ *is the pointwise extension of* $\subseteq$ *and the set of prototraces transformer* $\check{\mathsf{F}}^{\check{D}}[\![\mathcal{G}]\!] \in (\mathscr{N} \mapsto \wp(\Pi)) \mapsto (\mathscr{N} \mapsto \wp(\Pi))$ *is*

$$\check{\mathsf{F}}^{\check{D}}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, \phi \bullet \boldsymbol{\lambda}\, A \bullet \{\vdash \xrightarrow{\boxed{A}} \dashv\} \cup \mathrm{post}[\boxed{D} \Longrightarrow_{\mathcal{G}}] \phi(A) \ . \qquad \square$$

11 Abstraction of the Top-Down Protoderivation Semantics into the Bottom-Up Derivation Semantics

The trace derivations $\theta \in \mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!].A$ for a nonterminal A can be constructed top-down using the prototrace derivation $\boxed{D} \Longrightarrow_{\mathcal{G}}^{\star}$ as $(\vdash \xrightarrow{\boxed{A}} \dashv)\ \boxed{D} \Longrightarrow_{\mathcal{G}}^{\star} \theta$.

Lemma 13 *If* $T = \{\pi \in \Theta \mid \exists A \in \mathscr{N} : (\vdash \xrightarrow{\boxed{A}} \dashv)\ \boxed{D} \Longrightarrow_{\mathcal{G}}^{n*} \pi\}$ *then* $\hat{\check{\mathsf{F}}}^{\hat{d}}[A \to \sigma\centerdot\sigma'](T) = \{\pi \in \Theta \mid \check{\mathsf{F}}^{\check{D}}[A \to \sigma\centerdot\sigma'] \boxed{D} \Longrightarrow_{\mathcal{G}}^{n*} \pi\}$. $\square$

Lemma 14 *Let* $\hat{\check{\mathsf{F}}}^{\hat{d}}{}_n$ *be the iterates of* $\hat{\check{\mathsf{F}}}^{\hat{d}}[\![\mathcal{G}]\!]$ *from* $\hat{\check{\mathsf{F}}}^{\hat{d}}{}_0 = \varnothing$. *We have* $\hat{\check{\mathsf{F}}}^{\hat{d}}{}_n = \{\pi \in \Theta \mid \exists A \in \mathscr{N} : (\vdash \xrightarrow{\boxed{A}} \dashv)\ \boxed{D} \Longrightarrow_{\mathcal{G}}^{(n+1)*} \pi\}$ $\square$

Theorem 15 $\mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!] = \{\pi \in \Theta \mid \exists A \in \mathscr{N} : (\vdash \xrightarrow{\boxed{A}} \dashv)\ \boxed{D} \Longrightarrow_{\mathcal{G}}^{\star} \pi\}$. $\square$

Let us define the abstraction $\alpha^{\check{D}\hat{d}} \triangleq \boldsymbol{\lambda}\, P \bullet \boldsymbol{\lambda}\, A \bullet P(A) \cap \Theta$ which collects the terminal traces (without nonterminal variables) among prototraces. This abstraction defines a Galois connection [13] $\langle \mathscr{N} \mapsto \wp(\Pi),\ \dot{\subseteq} \rangle \underset{\alpha^{\check{D}\hat{d}}}{\overset{\gamma^{\check{D}\hat{d}}}{\leftrightarrows}} \langle \mathscr{N} \mapsto \wp(\Theta),\ \dot{\subseteq} \rangle$. The restriction of the top-down maximal protoderivation semantics is the maximal derivation semantics.

Theorem 16 $\alpha^{\check{D}\hat{d}}(\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!]) = \boldsymbol{\lambda}\, A \bullet \mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!].A$ . . $\square$

12 The Hierarchy of Grammar Semantics

Th. 16 shows that the bottom-up derivation semantics $\mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!]$ of a grammar $\mathcal{G}$ is, up to an isomorphism, an abstraction of the top-down protoderivation semantics $\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, A \bullet \{\pi \in \Pi \mid (\vdash \xrightarrow{\boxed{A}} \dashv)\ \boxed{D} \Longrightarrow_{\mathcal{G}}^{\star} \pi\}$ by the abstraction $\alpha^{\check{D}\hat{d}}$. We now introduce a hierarchy of abstractions of the protoderivation semantics $\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!]$, as given in **Fig. 1**. The various semantics and abstractions in **Fig. 1** (apart from $\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!]$, $\mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!]$, and $\alpha^{\check{D}\hat{d}}$) are described below.

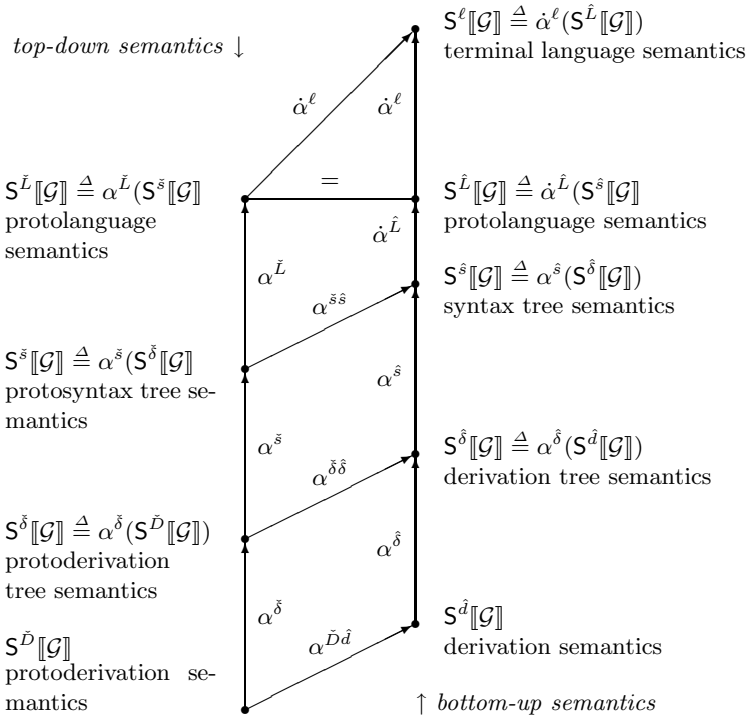
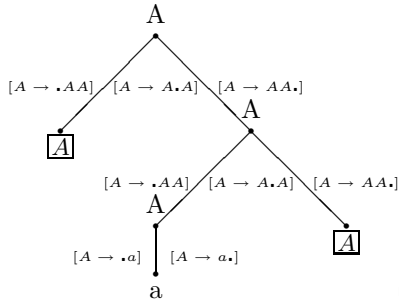


Fig. 1. The hierarchy of bottom-up grammar semantics

[Proto]derivation tree abstraction $\boldsymbol{\alpha^{\check{\delta}}}$ and $\boldsymbol{\alpha^{\hat{\delta}}}$. [Proto]derivations can be described by [proto]derivation trees where internal nodes are labelled with nonterminals, leafs are labelled with terminals [or nonterminal variables] and branches are decorated with rule states.

Example 17 One possible protoderivation tree for the protosentence AaA of the grammar $\langle\{a\}, \{A\}, A, \{A \to AA, A \to a\}\rangle$ is given on the right. It can be represented in parenthesized form through an infix traversal as $(\!|A[A \to \boldsymbol{.}AA]\ \boxed{A}\ [A \to A\boldsymbol{.}A](\!|A[A \to \boldsymbol{.}AA](\!|A[A \to \boldsymbol{.}a]a[A \to a\boldsymbol{.}]A|\!)[A \to A\boldsymbol{.}A]\boxed{A}[A \to AA\boldsymbol{.}]A|\!)[A \to AA\boldsymbol{.}]A|\!)$.



□

We let $\check{\mathscr{U}} \triangleq \mathscr{T} \cup \mathscr{N}^{\square} \cup \mathscr{R}^{\bullet}$ and $\check{\mathcal{D}} \triangleq (\mathscr{P} \cup \check{\mathscr{U}})^{\star}$. A *protoderivation tree* $\check{\delta}$ is represented by a well-parenthesized sentence over $\check{\mathscr{U}}$ so that $\check{\delta} \in \mathbb{P}_{\mathscr{P},\check{\mathscr{U}}} \subseteq \check{\mathcal{D}}$. We extend the selection to $\wp(\check{\mathcal{D}})$ whence $\wp(\mathbb{P}_{\mathscr{P},\check{\mathscr{U}}})$ as $D.A \triangleq \{(\!|B\sigma B|\!) \in D \mid B = A\} \cup \{\boxed{B} \in D \mid B = A\}$ so that $D.A$ is the set of protoderivation trees in D rooted at $A \in \mathscr{N}$.

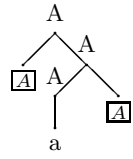
The *protoderivation tree abstraction* $\alpha^{\check{\delta}} \in \Pi \mapsto \check{\mathcal{D}}$ of protoderivations is

$$\begin{array}{lll} \alpha^{\check{\delta}}(\varpi \xrightarrow{\kappa} \tau) \triangleq \alpha^{\check{\delta}}(\varpi)\kappa\alpha^{\check{\delta}}(\tau) & & \alpha^{\check{\delta}}(\dashv) \triangleq \epsilon \\ \alpha^{\check{\delta}}(\epsilon) \triangleq \epsilon & & \alpha^{\check{\delta}}(s_1 \ldots s_n) \triangleq s_n, \quad s_1 \ldots s_n \in \mathcal{S}, \\ \alpha^{\check{\delta}}(\vdash) \triangleq \epsilon & & n > 0, \text{otherwise} \end{array}$$

which is extended elementwise to $\alpha^{\check{\delta}} \in \wp(\Pi) \mapsto \wp(\check{\mathcal{D}})$ as $\alpha^{\check{\delta}}(T) \triangleq \{\alpha^{\check{\delta}}(\pi) \mid \pi \in T\}$ so that we get the Galois connection $\langle \wp(\Pi), \subseteq\rangle \underset{\alpha^{\check{\delta}}}{\overset{\gamma^{\check{\delta}}}{\leftrightarrows}} \langle \wp(\check{\mathcal{D}}), \subseteq\rangle$, further extended pointwise to $\alpha^{\check{\delta}} \in (\mathscr{N} \mapsto \wp(\Pi)) \mapsto (\mathscr{N} \mapsto \wp(\check{\mathcal{D}}))$ as $\alpha^{\check{\delta}}(\phi) \triangleq \boldsymbol{\lambda} A \bullet \alpha^{\check{\delta}}(\phi(A))$. The restriction of $\alpha^{\check{\delta}}$ to derivation trees $\hat{\mathcal{D}} \triangleq (\mathscr{P} \cup \hat{\mathscr{U}})^\star$ where $\hat{\mathscr{U}} \triangleq \mathscr{T} \cup \mathscr{R}^\bullet$ is written $\alpha^{\hat{\delta}}$ so that $\langle \wp(\Theta), \subseteq\rangle \underset{\alpha^{\hat{\delta}}}{\overset{\gamma^{\hat{\delta}}}{\leftrightarrows}} \langle \wp(\hat{\mathcal{D}}), \subseteq\rangle$. A *derivation tree* $\hat{\delta}$ is represented by a well-parenthesized sentence over $\hat{\mathscr{U}}$ so that $\hat{\delta} \in \mathbb{P}_{\mathscr{P},\hat{\mathscr{U}}} \subseteq \hat{\mathcal{D}}$.

[Proto]syntax tree abstraction $\alpha^{\check{s}}$ and $\alpha^{\hat{s}}$. [Proto]syntax trees are [proto]-derivation trees denuded of the rule states decorating the branches. We represent [proto]syntax trees in parenthesized form through an infix traversal. We let $\check{\mathcal{T}} \triangleq (\mathscr{P} \cup \mathscr{T} \cup \mathscr{N}^\square)^\star$. A *protosyntax tree* $\check{\tau}$ is represented by a well-parenthesized sentence over $(\mathscr{T} \cup \mathscr{N}^\square)$ so that $\check{\tau} \in \mathbb{P}_{\mathscr{P},(\mathscr{T}\cup\mathscr{N}^\square)} \subseteq \check{\mathcal{T}}$.

Example 18 One possible protosyntax tree for the protosentence AaA of the grammar $\langle \{a\}, \{A\}, A, \{A \to AA, A \to a\}\rangle$ is given on the right and represented as $(\!|A\boxed{A}(\!|A(\!|AaA|\!)\boxed{A}A|\!)A|\!)$.



□

The *protosyntax tree abstraction* $\alpha^{\check{s}} \in \check{\mathcal{D}} \mapsto \check{\mathcal{T}}$ of protoderivation trees is ($A \in \mathscr{N}$, $\ell \in \mathscr{L}$)

$$\begin{array}{lll} \alpha^{\check{s}}(\sigma(\!|A\sigma') \triangleq \alpha^{\check{s}}(\sigma)(\!|A\alpha^{\check{s}}(\sigma') & & \alpha^{\check{s}}(\sigma[A \to \varsigma\bullet\varsigma']\sigma') \triangleq \alpha^{\check{s}}(\sigma)\alpha^{\check{s}}(\sigma') \\ \alpha^{\check{s}}(\sigma A|\!)\sigma') \triangleq \alpha^{\check{s}}(\sigma)A|\!)\alpha^{\check{s}}(\sigma') & & \alpha^{\check{s}}(\sigma\ell\sigma') \triangleq \alpha^{\check{s}}(\sigma)\ell\alpha^{\check{s}}(\sigma') \\ \alpha^{\check{s}}(\sigma\boxed{A}\sigma') \triangleq \alpha^{\check{s}}(\sigma)\boxed{A}\alpha^{\check{s}}(\sigma') & & \alpha^{\check{s}}(\epsilon) \triangleq \epsilon \end{array}$$

extended elementwise to $\alpha^{\check{s}} \in \wp(\check{\mathcal{D}}) \mapsto \wp(\check{\mathcal{T}})$ as $\alpha^{\check{s}}(D) \triangleq \{\alpha^{\check{s}}(\check{\delta}) \mid \check{\delta} \in D\}$ so that we get a Galois connection $\langle \wp(\check{\mathcal{D}}), \subseteq\rangle \underset{\alpha^{\check{s}}}{\overset{\gamma^{\check{s}}}{\leftrightarrows}} \langle \wp(\check{\mathcal{T}}), \subseteq\rangle$ which can be extended pointwise to $(\mathscr{N} \mapsto \wp(\check{\mathcal{D}})) \mapsto (\mathscr{N} \mapsto \wp(\check{\mathcal{T}}))$ as $\alpha^{\check{s}}(\phi) \triangleq \boldsymbol{\lambda} A \bullet \alpha^{\check{s}}(\phi(A))$. The restriction $\alpha^{\hat{s}}$ to syntax trees $\hat{\mathcal{T}} \triangleq (\mathscr{P} \cup \mathscr{T})^\star$ is such that $\langle \wp(\hat{\mathcal{D}}), \subseteq\rangle \underset{\alpha^{\hat{s}}}{\overset{\gamma^{\hat{s}}}{\leftrightarrows}} \langle \wp(\hat{\mathcal{T}}), \subseteq\rangle$. A *syntax tree* $\hat{\tau}$ is represented by a well-parenthesized sentence over $\mathscr{T}$ so that $\hat{\tau} \in \mathbb{P}_{\mathscr{P},\mathscr{T}} \subseteq \hat{\mathcal{T}}$.

Protosentence abstraction $\alpha^{\check{L}}$ and $\dot{\alpha}^{\hat{L}}$. The *protolanguage* of a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R}\rangle$ with $\mathscr{V} \triangleq \mathscr{T} \cup \mathscr{N}$ is the set of protosentences deriving from

the grammar axiom $\overline{S}$ where *protosentences* $\eta \in \mathscr{V}^\star$ contain both terminals in $\mathscr{T}$ and nonterminals in $\mathscr{N}$ and the derivation consists in replacing a nonterminal A by the righthand side σ of a grammar rule $A \to \sigma \in \mathscr{R}$.

The *protolanguage abstraction* $\alpha^{\check{L}} \in \check{\mathscr{T}} \mapsto \mathscr{V}^\star$ of protosyntax trees is defined as (we follow the tradition of confusing nonterminals A denoting the grammatical structure and nonterminal variables $\boxed{A}$ for protosentence substitution)

$$\begin{array}{lll}
\alpha^{\check{L}}(\sigma(\!|A\sigma'))\triangleq\alpha^{\check{L}}(\sigma)\alpha^{\check{L}}(\sigma'), \quad A\in\mathscr{N} & \alpha^{\check{L}}(\sigma a\sigma')\triangleq\alpha^{\check{L}}(\sigma)a\alpha^{\check{L}}(\sigma'), \quad a\in\mathscr{T} \\
\alpha^{\check{L}}(\sigma A|\!)\sigma')\triangleq\alpha^{\check{L}}(\sigma)\alpha^{\check{L}}(\sigma') & \alpha^{\check{L}}(\epsilon)\triangleq\epsilon \\
\alpha^{\check{L}}(\sigma\boxed{A}\sigma')\triangleq\alpha^{\check{L}}(\sigma)A\alpha^{\check{L}}(\sigma') &
\end{array}$$

extended elementwise to $\alpha^{\check{L}} \in \wp(\check{\mathscr{T}}) \mapsto \wp(\mathscr{V}^\star)$ as $\alpha^{\check{L}}(D) \triangleq \{\alpha^{\check{L}}(\check{\tau}) \mid \check{\tau} \in D\}$ so that we get a Galois connection $\langle \wp(\check{\mathscr{T}}), \subseteq\rangle \underset{\alpha^{\check{L}}}{\overset{\gamma^{\check{L}}}{\leftrightarrows}} \langle \wp(\mathscr{V}^\star), \subseteq\rangle$ which can be extended pointwise to $\alpha^{\check{L}} \in (\mathscr{N} \mapsto \wp(\check{\mathscr{T}})) \mapsto (\mathscr{N} \mapsto \wp(\mathscr{V}^\star))$ as $\alpha^{\check{L}}(\phi) \triangleq \boldsymbol{\lambda}\, A \bullet \alpha^{\check{L}}(\phi(A))$.

Example 19 For the protosyntax tree in **Ex. 18** of the grammar $\langle\{a\}, \{A\}, A, \{A \to AA, A \to a\}\rangle$, we have $\alpha^{\check{L}}\big(\!(A\boxed{A}\!(A\!(AaA|\!)\boxed{A}A|\!)A|\!)\big) = AaA$. □

For syntax trees, we define the flattener $\alpha^{\hat{L}} \in \hat{\mathscr{T}} \mapsto \wp(\mathscr{V}^\star)$ as

$$\alpha^{\hat{L}}(\!(A\sigma A|\!)\sigma') \triangleq (\{A\} \cup \alpha^{\hat{L}}(\sigma))\alpha^{\hat{L}}(\sigma') \qquad \alpha^{\hat{L}}(a\sigma') \triangleq \{a\}\alpha^{\hat{L}}(\sigma') \qquad \alpha^{\hat{L}}(\epsilon) \triangleq \{\epsilon\}$$

extended elementwise to $\alpha^{\hat{L}} \in \wp(\hat{\mathscr{T}}) \mapsto \wp(\mathscr{V}^\star)$ as $\alpha^{\hat{L}}(\Sigma) \triangleq \bigcup\{\alpha^{\hat{L}}(\sigma) \mid \sigma \in \Sigma\}$ and pointwise to $\dot{\alpha}^{\hat{L}} \in \wp(\hat{\mathscr{T}}) \mapsto (\mathscr{N} \mapsto \wp(\mathscr{V}^\star))$ as $\dot{\alpha}^{\hat{L}}(S) \triangleq \boldsymbol{\lambda}\, A \bullet \alpha^{\hat{L}}(S.A)$ so that we get the Galois connection $\langle \wp(\hat{\mathscr{T}}), \subseteq\rangle \underset{\dot{\alpha}^{\hat{L}}}{\overset{\dot{\gamma}^{\hat{L}}}{\leftrightarrows}} \langle \mathscr{N} \mapsto \wp(\mathscr{V}^\star), \dot{\subseteq}\rangle$.

Terminal sentence abstraction $\dot{\alpha}^{\ell}$. Terminal sentence abstraction eliminates the sentences of a protolanguage which are not terminal. Let us define the eraser $\alpha^{\ell} \in \mathscr{V}^\star \mapsto \wp(\mathscr{T}^\star)$ as

$$\alpha^{\ell}(A\sigma) \triangleq \varnothing \qquad \alpha^{\ell}(a\sigma) \triangleq a\alpha^{\ell}(\sigma) \qquad \alpha^{\ell}(\epsilon) \triangleq \epsilon$$

extended to $\alpha^{\ell} \in \wp(\mathscr{V}^\star) \mapsto \wp(\mathscr{T}^\star)$ as $\alpha^{\ell}(\Sigma) \triangleq \bigcup\{\alpha^{\ell}(\sigma) \mid \sigma \in \Sigma\} = \Sigma \cap \mathscr{T}^\star$ so that we get a Galois connection $\langle \wp(\mathscr{V}^\star), \subseteq\rangle \underset{\alpha^{\ell}}{\overset{\gamma^{\ell}}{\leftrightarrows}} \langle \wp(\mathscr{T}^\star), \subseteq\rangle$ which can be extended pointwise to $\dot{\alpha}^{\ell} \in (\mathscr{N} \mapsto \wp(\mathscr{V}^\star)) \mapsto (\mathscr{N} \mapsto \wp(\mathscr{T}^\star))$ as $\dot{\alpha}^{\ell}(\rho) \triangleq \boldsymbol{\lambda}\, A \bullet \alpha^{\ell}(\rho(A))$.

13 Fixpoint Bottom-Up Abstract Semantics

All bottom-up semantics $\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!] \in \hat{\mathsf{D}}^{\hat{\sharp}}$ of context-free grammars $\mathcal{G}$ are instances of the following abstract interpreter (which generalizes the bottom-up grammar

flow analysis of [8, Def. 8.2.18]).

$$\mathsf{S}^{\hat\sharp}[\![\mathcal{G}]\!] = \mathbf{lfp}^{\sqsubseteq}\, \hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[\![\mathcal{G}]\!] \tag{13}$$

where $\langle \hat{\mathsf{D}}^{\hat\sharp}, \sqsubseteq, \bot, \sqcup\rangle$ is a cpo/complete lattice and the transformer $\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[\![\mathcal{G}]\!] \in \hat{\mathsf{D}}^{\hat\sharp} \mapsto \hat{\mathsf{D}}^{\hat\sharp}$ is

$$\begin{aligned}
\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[\![\mathcal{G}]\!] &\triangleq \boldsymbol{\lambda}\, \rho \bullet \bigsqcup_{A\to\sigma\in\mathscr{R}} A^{\hat\sharp}(\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \mathbf{.}\sigma]\rho) \qquad (14)\\
\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \sigma\mathbf{.}a\sigma'] &\triangleq \boldsymbol{\lambda}\, \rho \bullet [A \to \sigma\mathbf{.}a\sigma']^{\hat\sharp} \mathbin{\hat\circ^{\hat\sharp}} \hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \sigma a\mathbf{.}\sigma']\rho\\
\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \sigma\mathbf{.}B\sigma'] &\triangleq \boldsymbol{\lambda}\, \rho \bullet [A \to \sigma\mathbf{.}B\sigma']^{\hat\sharp}(\rho, B) \mathbin{\hat\wr^{\hat\sharp}} \hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \sigma B\mathbf{.}\sigma']\rho\\
\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \sigma\mathbf{.}] &\triangleq \boldsymbol{\lambda}\, \rho \bullet [A \to \sigma\mathbf{.}]^{\hat\sharp}
\end{aligned}$$

where the abstract rooting is $A^{\hat\sharp} \in \hat{\mathsf{D}}^{\hat\sharp} \mapsto \hat{\mathsf{D}}^{\hat\sharp}$, $[A \to \sigma\mathbf{.}a\sigma']^{\hat\sharp} \in \hat{\mathsf{D}}^{\hat\sharp}$, the abstract concatenation is $\hat\circ^{\hat\sharp} \in (\hat{\mathsf{D}}^{\hat\sharp} \times \hat{\mathsf{D}}^{\hat\sharp}) \mapsto \hat{\mathsf{D}}^{\hat\sharp}$, $[A \to \sigma\mathbf{.}B\sigma']^{\hat\sharp} \in (\hat{\mathsf{D}}^{\hat\sharp} \times \mathscr{N}) \mapsto \hat{\mathsf{D}}^{\hat\sharp}$, the abstract junction is $\hat\wr^{\hat\sharp} \in (\hat{\mathsf{D}}^{\hat\sharp} \times \hat{\mathsf{D}}^{\hat\sharp}) \mapsto \hat{\mathsf{D}}^{\hat\sharp}$, and $[A \to \sigma\mathbf{.}]^{\hat\sharp} \in \hat{\mathsf{D}}^{\hat\sharp}$.

The existence of the least fixpoint is guaranteed by the following

Hypothesis 20 *For all* $[A \to \sigma\mathbf{.}\sigma'] \in \mathscr{R}^{\cdot}$, $\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \sigma\mathbf{.}\sigma'] \in (\mathscr{N} \mapsto L) \longmapsto L$ *is upper continuous for the ordering* $\sqsubseteq$ *on* $\hat{\mathsf{D}}^{\hat\sharp}$[3]. □

Hyp. 20 is guaranteed by the following local continuity conditions

Lemma 21 *If* $A^{\hat\sharp}$ *is continuous,* $\hat\circ^{\hat\sharp}$ *is continuous in its second argument,* $[A \to \sigma\mathbf{.}B\sigma']^{\hat\sharp}$ *is continuous in its first argument,* $\hat\wr^{\hat\sharp}$ *is continuous then* ***Hyp. 20*** *holds.* □

The hierarchy of semantics discussed in **Sect. 12** is obtained by the instances of the bottom-up abstract semantics (13) given in **Fig. 2**. Classical semantics and flow analyzes also have the same form given in **Fig. 3** (where $\mathbb{B} \triangleq \{\mathtt{ff}, \mathtt{tt}\}$).

We can define the soundness of an abstract interpreter $\mathsf{S}^{\hat\sharp}[\![\mathcal{G}]\!]$ with respect to a concrete interpreter $\mathsf{S}^{\natural}[\![\mathcal{G}]\!]$ as $\alpha(\mathsf{S}^{\natural}[\![\mathcal{G}]\!]) = \mathsf{S}^{\hat\sharp}[\![\mathcal{G}]\!]$ using a Galois connection $\langle L^{\natural}, \sqsubseteq^{\natural}\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L^{\hat\sharp}, \sqsubseteq^{\hat\sharp}\rangle$. This global soundness condition on the abstraction is implied by the *rule soundness condition*

$$\alpha(A^{\natural}(\hat{\overrightarrow{\mathsf{F}}}^{\natural}[A \to \mathbf{.}\sigma]\rho)) = A^{\hat\sharp}(\hat{\overrightarrow{\mathsf{F}}}^{\hat\sharp}[A \to \mathbf{.}\sigma]\alpha(\rho)) \tag{15}$$

which is itself implied by the *local soundness conditions* on the abstract operators (for all $x, y, \rho \in L^{\hat\sharp}$)

$$\begin{aligned}
\alpha(A^{\natural}(x)) &= A^{\hat\sharp}(\alpha(x)), & \alpha([A \to \sigma\mathbf{.}B\sigma']^{\natural}(\rho, B)) &= [A \to \sigma\mathbf{.}B\sigma']^{\hat\sharp}(\alpha(\rho), B),\\
\alpha([A \to \sigma\mathbf{.}a\sigma']^{\natural}) &= [A \to \sigma\mathbf{.}a\sigma']^{\hat\sharp}, & \alpha(x \mathbin{\wr^{\natural}} y) &= \alpha(x) \mathbin{\hat\wr^{\hat\sharp}} \alpha(y),\\
\alpha(x \mathbin{\circ^{\natural}} y) &= \alpha(x) \mathbin{\hat\circ^{\hat\sharp}} \alpha(y), & \alpha([A \to \sigma\mathbf{.}]^{\natural}) &= [A \to \sigma\mathbf{.}]^{\hat\sharp}\ .
\end{aligned}$$

[3] Indeed monotony is sufficient [14].

Abstract semantics $\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!]$	Maximal derivation $\mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!]$	Derivation tree $\mathsf{S}^{\hat{s}}[\![\mathcal{G}]\!]$	Syntax tree $\mathsf{S}^{\hat{s}}[\![\mathcal{G}]\!]$	Proto-language $\mathsf{S}^{\hat{L}}[\![\mathcal{G}]\!]$
$\hat{\mathsf{D}}^{\hat{\sharp}}$	$\wp(\Theta)$	$\wp(\hat{\mathcal{D}})$	$\wp(\hat{\mathcal{T}})$	$\mathscr{N} \mapsto \wp(\mathscr{V}^\star)$
$\sqsubseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\dot{\subseteq}$
$\bot$	$\varnothing$	$\varnothing$	$\varnothing$	$\dot{\varnothing}$
$\sqcup$	$\cup$	$\cup$	$\cup$	$\dot{\cup}$
$A^{\hat{\sharp}}(X)$	$\vdash \xrightarrow{(\!\lvert A} X \xrightarrow{A\rvert\!)} \dashv$	$(\!\lvert AXA \rvert\!)$	$(\!\lvert AXA \rvert\!)$	$A^{\hat{L}}(X)$
$[A \to \sigma\centerdot a\sigma']^{\hat{\sharp}}$	$(\dashv[A \to \sigma\centerdot a\sigma']) \xrightarrow{a}$	$[A \to \sigma\centerdot a\sigma']a$	a[4]	$\boldsymbol{\lambda}\, A' \centerdot a$
$\circ^{\hat{\sharp}}$	$\dot{\cdot}$[5]	$\dot{\cdot}$	$\dot{\cdot}$	$\dot{\cdot}$
$[A \to \sigma\centerdot B\sigma']^{\hat{\sharp}}(\rho, B)$	$[A \to \sigma\centerdot B\sigma']^{\hat{d}}(\rho, B)$	$[A \to \sigma\centerdot B\sigma']\ \rho.B$	$\rho.B$	$\boldsymbol{\lambda}\, A' \centerdot \{B\} \cup \rho(B)$
$\wr^{\hat{\sharp}}$	$\wr$	$\dot{\cdot}$	$\dot{\cdot}$	$\dot{\cdot}$
$[A \to \sigma\centerdot]^{\hat{\sharp}}$	$\dashv[A \to \sigma\centerdot]$	$[A \to \sigma\centerdot]$	ϵ	$\boldsymbol{\lambda}\, A' \centerdot \epsilon$

where $A^{\hat{L}}(X) \triangleq \boldsymbol{\lambda}\, A' \centerdot (\!\lvert A' = A \mathbin{?} \{A\} \cup X(A) \mathbin{:} \varnothing \rvert\!)$[6] and $[A \to \sigma\centerdot B\sigma']^{\hat{d}}(\rho, B) \triangleq \langle \dashv[A \to \sigma\centerdot B\sigma'],\ \dashv[A \to \sigma B\centerdot \sigma'] \rangle \uparrow \rho.B$.

Fig. 2. Semantic instances of the abstract bottom-up grammar semantics (13)

Theorem 22 *The above local soundness conditions imply the soundness and completeness of the abstract interpreter* $\alpha(\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!]) = \mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!]$. □

For example, the terminal language semantics $\mathsf{S}^{\ell}[\![\mathcal{G}]\!]$ defines the classical equational definition of the language generated by a grammar [15,16].

Theorem 23 (Ginsburg, Rice, Schützenberger) $\mathsf{S}^{\ell}[\![\mathcal{G}]\!] = \mathit{lfp}^{\dot{\subseteq}}\, \vec{\hat{\mathsf{F}}}^{\ell}[\![\mathcal{G}]\!]$. □

Example 24 For the grammar $\mathcal{G} = \langle \{(,)\},\ \{A\},\ A,\ \{A \to (A)A, A \to \epsilon\} \rangle$, the fixpoint equation $\rho = \vec{\hat{\mathsf{F}}}^{\ell}[\![\mathcal{G}]\!](\rho)$ or equivalently $\rho(A) = \vec{\hat{\mathsf{F}}}^{\ell}[\![\mathcal{G}]\!](\rho)(A)$ is $\rho(A) = (\rho(A))\rho(A) \cup \epsilon$, which defining $\mathcal{X} = \rho(A)$, is $\mathcal{X} = \{(\}\mathcal{X}\{)\}\mathcal{X} \cup \{\epsilon\}$ which generates the Dyck language over parentheses $\{(,)\}$ that is, by iteration, $\{\epsilon\} \cup \{()\} \cup \{(()), ()()\} \cup \ldots$. □

14 Extension of the Bottom-Up Structural Abstract Semantics to Grammar Rule States

When $\hat{\mathsf{D}}^{\hat{\sharp}}$ is of the form $\mathscr{N} \mapsto L$, the abstract semantics $\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!] \in \mathscr{N} \mapsto L$ can be extended to grammar rule states $\vec{\hat{\mathsf{S}}}^{\hat{\sharp}}[\![\mathcal{G}]\!] \in \mathscr{R}^{\centerdot} \mapsto L$ as

$$\vec{\hat{\mathsf{S}}}^{\hat{\sharp}}[\![\mathcal{G}]\!][A \to \sigma\centerdot\sigma'] \triangleq \vec{\hat{\mathsf{F}}}^{\hat{\sharp}}[A \to \sigma\centerdot\sigma'](\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!]) \tag{16}$$

where $\vec{\hat{\mathsf{F}}}^{\hat{\sharp}}[\![\mathcal{G}]\!] \in (\mathscr{R}^{\centerdot} \mapsto L) \mapsto (\mathscr{R}^{\centerdot} \mapsto L)$ is

$$\vec{\hat{\mathsf{F}}}^{\hat{\sharp}}[\![\mathcal{G}]\!]\rho[A \to \sigma\centerdot a\sigma'] \triangleq [A \to \sigma\centerdot a\sigma']^{\hat{\sharp}} \mathbin{\circ^{\hat{\sharp}}} \vec{\hat{\mathsf{F}}}^{\hat{\sharp}}[\![\mathcal{G}]\!]\rho[A \to \sigma a\centerdot\sigma'] \tag{17}$$

Abstract semantics $\mathsf{S}^{\hat\sharp}[\![\mathcal{G}]\!]$	Terminal language $\mathsf{S}^{\ell}[\![\mathcal{G}]\!]$	First $\mathsf{S}^{1}[\![\mathcal{G}]\!]$	ϵ−Productivity $\mathsf{S}^{\epsilon}[\![\mathcal{G}]\!]$	Nonterminal productivity $\mathsf{S}^{\otimes}[\![\mathcal{G}]\!]$
$\hat{\mathsf{D}}^{\hat\sharp}$	$\mathscr{N} \mapsto \wp(\mathscr{T}^\star)$	$\mathscr{N} \mapsto \wp(\mathscr{T} \cup \{\epsilon\})$	$\mathscr{N} \mapsto \mathbb{B}$	$\mathscr{N} \mapsto \mathbb{B}$
$\sqsubseteq$	$\dot\subseteq$	$\dot\subseteq$	$\dot\Longrightarrow$	$\dot\Longrightarrow$
$\bot$	$\dot\varnothing$	$\dot\varnothing$	$\boldsymbol{\lambda}\, N \cdot \mathtt{ff}$	$\boldsymbol{\lambda}\, N \cdot \mathtt{ff}$
$\sqcup$	$\dot\cup$	$\dot\cup$	$\dot\vee$	$\dot\vee$
$A^{\hat\sharp}(X)$	$A^{\ell}(X)$	$A^{1}(X)$	$A^{\epsilon}(X)$	$A^{\otimes}(X)$
$[A \to \sigma\mathbf{.}a\sigma']^{\hat\sharp}$	$\boldsymbol{\lambda}\, A' \cdot a$	$\boldsymbol{\lambda}\, A' \cdot a$	$\boldsymbol{\lambda}\, A' \cdot \mathtt{ff}$	$\boldsymbol{\lambda}\, A' \cdot \mathtt{tt}$
$\circ^{\hat\sharp}$	$.$	$\dot\oplus^{1}$	$\dot\wedge$	$\dot\wedge$
$[A \to \sigma\mathbf{.}B\sigma']^{\hat\sharp}(\rho, B)$	$\boldsymbol{\lambda}\, A' \cdot \rho(B)$	$\boldsymbol{\lambda}\, A' \cdot \rho(B)$	$\boldsymbol{\lambda}\, A' \cdot \rho(B)$	$\boldsymbol{\lambda}\, A' \cdot \rho(B)$
$\wr^{\hat\sharp}$	$.$	$\dot\oplus^{1}$	$\dot\wedge$	$\dot\wedge$
$[A \to \sigma\mathbf{.}]^{\hat\sharp}$	$\boldsymbol{\lambda}\, A' \cdot \epsilon$	$\boldsymbol{\lambda}\, A' \cdot \epsilon$	$\boldsymbol{\lambda}\, A' \cdot \mathtt{tt}$	$\boldsymbol{\lambda}\, A' \cdot \mathtt{tt}$

where $A^{\ell}(X) = A^{1}(X) \triangleq \boldsymbol{\lambda}\, A' \cdot (\!|A' = A \mathbin{?} X(A) \mathbin{:} \varnothing|\!)$, $A^{\epsilon}(X) = A^{\otimes}(X) \triangleq \boldsymbol{\lambda}\, A' \cdot (\!|A' = A \mathbin{?} X(A) \mathbin{:} \mathtt{ff}|\!)$, the first abstraction $\oplus^{1}$ of language concatenation is defined in **Lem. 29**, and $\dot\oplus^{1}$ is its pointwise extension.

Fig. 3. Flow analysis instances of the abstract bottom-up grammar semantics (13)

$$\overset{\hat\to}{\mathsf{F}}{}^{\hat\sharp}[\![\mathcal{G}]\!]\rho[A \to \sigma\mathbf{.}B\sigma'] \triangleq [A \to \sigma\mathbf{.}B\sigma']^{\hat\sharp}\Big(\bigsqcup_{C \to \varsigma \in \mathscr{R}} C^{\hat\sharp}(\overset{\hat\to}{\mathsf{F}}{}^{\hat\sharp}[C \to \mathbf{.}\varsigma](\rho)), B\Big) \wr^{\hat\sharp} \overset{\hat\to}{\mathsf{F}}{}^{\hat\sharp}[\![\mathcal{G}]\!]\rho[A \to \sigma B\mathbf{.}\sigma']$$

$$\overset{\hat\to}{\mathsf{F}}{}^{\hat\sharp}[\![\mathcal{G}]\!]\rho[A \to \sigma\mathbf{.}] \triangleq [A \to \sigma\mathbf{.}]^{\hat\sharp}$$

with the following fixpoint characterization

Theorem 25

$$\overset{\hat\to}{\mathsf{S}}{}^{\hat\sharp}[\![\mathcal{G}]\!] = \mathit{lfp}^{\sqsubseteq}\, \overset{\hat\to}{\mathsf{F}}{}^{\hat\sharp}[\![\mathcal{G}]\!] \ . \qquad \square$$

The relationship between the abstract semantics $\mathsf{S}^{\hat\sharp}[\![\mathcal{G}]\!]$ and its extension $\overset{\hat\to}{\mathsf{S}}{}^{\hat\sharp}[\![\mathcal{G}]\!]$ to grammar rule states is given by (16) and the following

Theorem 26 *If $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ is a grammar then*

$$\mathsf{S}^{\hat\sharp}[\![\mathcal{G}]\!] = \bigsqcup_{A \to \sigma \in \mathscr{R}} A^{\hat\sharp}(\overset{\hat\to}{\mathsf{S}}{}^{\hat\sharp}[\![\mathcal{G}]\!][A \to \mathbf{.}\sigma]) \ . \qquad \square$$

15 Fixpoint Top-Down Abstract Semantics

The top-down semantics in the hierarchy of **Sect. 12** can all be viewed as instances of an abstract interpreter generalizing the top-down flow analysis of [8, Def. 8.2.19]. For brevity, we consider only the *protolanguage semantics* $\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!] \in \mathscr{N} \mapsto \wp(\mathscr{V}^\star)$ of a context-free grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$, which is the protolanguage generated by the grammar $\mathcal{G}$ for each nonterminal. It is defined as

$$\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!] \triangleq \alpha^{\check{L}}(\alpha^{\check{s}}(\alpha^{\check{\delta}}(\mathsf{S}^{\check{D}}[\![\mathcal{G}]\!]))) \ . \tag{18}$$

Let us define the *protolanguage derivation* $\Longmapsto_{\mathcal{G}}$ for a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ ($\Longmapsto$ when $\mathcal{G}$ is understood)

$$\begin{aligned} &\eta \Longmapsto_{\mathcal{G}} \eta' \\ &\triangleq \exists n > 0, \varsigma_1, \ldots, \varsigma_{n+1}, A_1, \ldots, A_n, \sigma_1, \ldots, \sigma_n : \eta = \varsigma_1 A_1 \varsigma_2 \ldots \varsigma_n A_n \varsigma_{n+1} \wedge \\ &\qquad \forall i \in [1, n] : A_i \to \sigma_i \in \mathscr{R} \wedge \eta' = \varsigma_1 \sigma_1 \varsigma_2 \ldots \varsigma_n \sigma_n \varsigma_{n+1} \ . \end{aligned} \tag{19}$$

This is [8, Def. 8.2.2] for $n = 1$, the difference being that we allow several simultaneous substitutions.

The protolanguage semantics can be defined in fixpoint form as

Theorem 27

$$\begin{aligned} &\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!] = \mathit{lfp}^{\dot{\subseteq}} \check{\mathsf{F}}^{\check{L}}[\![\mathcal{G}]\!] \\ &\textit{where} \quad \check{\mathsf{F}}^{\check{L}}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, \phi \bullet \boldsymbol{\lambda}\, A \bullet \{A\} \cup \mathrm{post}[\Longmapsto_{\mathcal{G}}]\phi(A) \ . \end{aligned}$$

□

As a corollary of this proof and (12), it follows that

$$\boldsymbol{\lambda}\, A \bullet \{\alpha^{\check{L}}(\alpha^{\check{s}}(\alpha^{\check{\delta}}(\pi))) \mid (\vdash \xrightarrow{\boxed{A}} \dashv) \ \mathbb{D}\overset{\star}{\Longmapsto}_{\mathcal{G}} \pi\} = \boldsymbol{\lambda}\, A \bullet \{\eta \mid A \Longmapsto_{\mathcal{G}} \eta\} \tag{20}$$

so that we also have the classical definition of the protolanguage generated by a grammar [8, Def. 8.2.3]

$$\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!] = \boldsymbol{\lambda}\, A \bullet \{\eta \in \mathscr{V}^\star \mid A \overset{\star}{\Longmapsto}_{\mathcal{G}} \eta\} \ . \tag{21}$$

Applying the terminal language abstraction, we get the classical definition of the terminal language generated by a grammar [8, Def. 8.2.3]

Theorem 28 $\mathsf{S}^{\ell}[\![\mathcal{G}]\!] \triangleq \dot{\alpha}^{\ell}(\mathsf{S}^{\hat{L}}[\![\mathcal{G}]\!]) = \boldsymbol{\lambda}\, A \bullet \{\sigma \in \mathscr{T}^\star \mid A \overset{\star}{\Longmapsto}_{\mathcal{G}} \sigma\}$. □

The protolanguage semantics $\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!] \in \mathscr{N} \mapsto \wp(\mathscr{V}^\star)$ can be extended to grammar rule states $\overset{\rightarrow}{\mathsf{S}}{}^{\check{L}}[\![\mathcal{G}]\!] \in \mathscr{R}^\bullet \mapsto \wp(\mathscr{V}^\star)$ as follows

$$\begin{aligned} \overset{\rightarrow}{\mathsf{S}}{}^{\check{L}}[\![\mathcal{G}]\!][A \to \sigma \bullet a \sigma'] &\triangleq a \, \overset{\rightarrow}{\mathsf{S}}{}^{\check{L}}[\![\mathcal{G}]\!][A \to \sigma a \bullet \sigma'] \\ \overset{\rightarrow}{\mathsf{S}}{}^{\check{L}}[\![\mathcal{G}]\!][A \to \sigma \bullet B \sigma'] &\triangleq \mathsf{S}^{\check{L}}[\![\mathcal{G}]\!](B) \, \overset{\rightarrow}{\mathsf{S}}{}^{\check{L}}[\![\mathcal{G}]\!][A \to \sigma B \bullet \sigma'] \\ \overset{\rightarrow}{\mathsf{S}}{}^{\check{L}}[\![\mathcal{G}]\!][A \to \sigma \bullet] &\triangleq \epsilon \end{aligned} \tag{22}$$

so that

$$\overset{\rightarrow}{\mathsf{S}}{}^{\check{L}}[\![\mathcal{G}]\!][A \to \sigma \bullet \sigma'] = \{\varsigma \in \mathscr{V}^\star \mid \sigma' \overset{\star}{\Longmapsto}_{\mathcal{G}} \varsigma\} \ . \tag{23}$$

16 Bottom-Up Grammar Analysis

Classical grammar analysis algorithms such as FIRST [8, Sect. 8.2.8], nonterminal productivity [8, Sect. 8.2.4], and ϵ-productivity ϵ-PROD [8, Sect. 8.2.3] are abstractions of the bottom-up grammar semantics and are instances of the bottom-up abstract interpreter (13).

16.1 First

The *first abstraction* $\alpha^1 \in \mathscr{T}^\star \mapsto \wp(\mathscr{T} \cup \{\epsilon\})$ of a terminal sentence is the first terminal of this sentence or ϵ for empty sentences. $\alpha^1 \triangleq \boldsymbol{\lambda}\, \sigma \bullet \{a \in \mathscr{T} \mid \exists \sigma' \in \mathscr{T}^\star : \sigma = a\sigma'\} \cup \{\epsilon \mid \sigma = \epsilon\}$. It is extended to terminal languages $\alpha^1 \in \wp(\mathscr{T}^\star) \mapsto \wp(\mathscr{T} \cup \{\epsilon\})$ in order to collect the first terminals of the sentences of these languages $\alpha^1 \triangleq \boldsymbol{\lambda}\, \Sigma \bullet \bigcup_{\sigma \in \Sigma} \alpha^1(\sigma)$ and finally extended pointwise $\dot{\alpha}^1 \in (\mathscr{N} \mapsto \wp(\mathscr{T}^\star)) \mapsto (\mathscr{N} \mapsto \wp(\mathscr{T} \cup \{\epsilon\}))$ on terminal languages derived for nonterminals as $\dot{\alpha}^1 \triangleq \boldsymbol{\lambda}\, L \bullet \boldsymbol{\lambda}\, A \bullet \alpha^1(L(A))$.

The first abstraction of language concatenation is

Lemma 29 *For all $\Sigma, \Sigma' \in \wp(\mathscr{T}^\star)$ and $F, F' \in \wp(\mathscr{T})$,*

$$\begin{array}{ll} & \alpha^1(\Sigma\Sigma') = \alpha^1(\Sigma) \oplus^1 \alpha^1(\Sigma') \\ \textit{where} & F \oplus^1 F' \triangleq \left(F' \neq \varnothing \mathbin{?} (F \setminus \{\epsilon\}) \cup \left(\epsilon \in F \mathbin{?} F' \mathbin{:} \varnothing\right) \mathbin{:} \varnothing\right) \\ \textit{and} & \{a\} \oplus^1 F' \triangleq \left(F' \neq \varnothing \mathbin{?} \{a\} \mathbin{:} \varnothing\right) . \end{array}$$ □

The first concatenation is monotone (hence upper-continuous since $\mathscr{T}$ is finite)

Lemma 30 *If $F_1 \subseteq F_1'$ and $F_2 \subseteq F_2'$ then $F_1 \oplus^1 F_2 \subseteq F_1' \oplus^1 F_2'$.* □

The *first semantics* $\mathsf{S}^1[\![\mathcal{G}]\!] \in \mathscr{N} \mapsto \wp(\mathscr{T} \cup \{\epsilon\})$ of a grammar $\mathcal{G}$ is

$$\mathsf{S}^1[\![\mathcal{G}]\!] \triangleq \dot{\alpha}^1(\mathsf{S}^\ell[\![\mathcal{G}]\!]) \ . \tag{24}$$

The classical definition of the FIRST derivation of a grammar [8, Def. 8.2.33] is

Theorem 31

$$\mathsf{S}^1[\![\mathcal{G}]\!] = \boldsymbol{\lambda}\, A \bullet \{a \in \mathscr{T} \mid \exists \sigma \in \mathscr{T}^\star : A \overset{\star}{\Longmapsto}_{\mathcal{G}} a\sigma\} \cup \{\epsilon \mid A \overset{\star}{\Longmapsto}_{\mathcal{G}} \epsilon\} \ .$$ □

For parsing, the input sentence is often assumed to be followed by the final mark $\dashv$, so it is useful to extend $\mathsf{S}^1[\![\mathcal{G}]\!]$ to $\mathsf{S}^{1\dashv}[\![\mathcal{G}]\!] \in \mathscr{N} \mapsto \wp(\mathscr{T} \cup \{\dashv\})$ as

$$\mathsf{S}^{1\dashv}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, A \bullet \{a \in \mathscr{T} \mid \exists \sigma \in \mathscr{T}^\star : A \overset{\star}{\Longmapsto}_{\mathcal{G}} a\sigma\} \cup \{\dashv \mid A \overset{\star}{\Longmapsto}_{\mathcal{G}} \epsilon\} \ . \tag{25}$$

The FIRST algorithm [8, **Fig. 8.11**] is indeed a fixpoint computation $\mathsf{S}^1[\![\mathcal{G}]\!] = \mathbf{lfp}^{\subseteq} \hat{\overrightarrow{\mathsf{F}}}^1[\![\mathcal{G}]\!]$ where the bottom-up transformer $\hat{\overrightarrow{\mathsf{F}}}^1[\![\mathcal{G}]\!]$ is (14) instantiated as given in **Sect. 13**[7].

[7] The classical definition [8, **Fig. 8.11**] is simpler since all grammar nonterminals are assumed to be productive.

16.2 ϵ-Productivity

The classical definition of ϵ-PROD [8, Sect. 8.2.3] provides information on which nonterminals can be empty. The corresponding abstraction is $\alpha^\epsilon \triangleq \boldsymbol{\lambda}\, \Sigma \bullet \left(\epsilon \in \Sigma \;?\; \mathrm{tt} \;:\; \mathrm{ff}\right)$ extended pointwise to $\alpha^\epsilon \triangleq \boldsymbol{\lambda}\, L \bullet \boldsymbol{\lambda}\, A \bullet \alpha^\epsilon(L(A))$ so that $\langle \mathscr{N} \mapsto \wp(\mathscr{T}^\star),\ \dot{\subseteq} \rangle \underset{\alpha^\epsilon}{\overset{\dot{\gamma}^\epsilon}{\leftrightarrows}} \langle \mathscr{N} \mapsto \mathbb{B},\ \dot{\Longrightarrow} \rangle$. The *$\epsilon$-productivity semantics* $\mathsf{S}^\epsilon[\![\mathcal{G}]\!] \triangleq \alpha^\epsilon(\mathsf{S}^\ell[\![\mathcal{G}]\!]) = \alpha^\epsilon(\mathsf{S}^1[\![\mathcal{G}]\!])$ since $\alpha^\epsilon = \alpha^\epsilon \circ \dot{\alpha}^1$ and $\mathsf{S}^1[\![\mathcal{G}]\!] = \dot{\alpha}^1(\mathsf{S}^\ell[\![\mathcal{G}]\!])$. This is the classical definition of ϵ-productivity for a grammar [8, Sect. 8.2.9] since $\mathsf{S}^\epsilon[\![\mathcal{G}]\!] = \boldsymbol{\lambda}\, A \bullet A \overset{\star}{\Longmapsto}_{\mathcal{G}} \epsilon$. The ϵ-PRODUCTIVITY iterative computation [8, **Fig. 8.14**] is indeed a fixpoint computation $\mathsf{S}^\epsilon[\![\mathcal{G}]\!] = \mathbf{lfp}^{\dot{\Longrightarrow}}\, \hat{\dot{\mathsf{F}}}^\epsilon[\![\mathcal{G}]\!]$ where the bottom-up transformer $\hat{\dot{\mathsf{F}}}^\epsilon[\![\mathcal{G}]\!]$ is (14) instantiated as given in **Sect. 13**.

16.3 Nonterminal Productivity

The classical definition of *nonterminal productivity* [8, Sect. 8.2.4] provides information on which nonterminals of the grammar can produce a non-empty terminal language. The nonterminal productivity semantics of a context-free grammar is indeed an abstraction of its first semantics $\mathsf{S}^\otimes[\![\mathcal{G}]\!] \triangleq \dot{\alpha}^\otimes(\mathsf{S}^\ell[\![\mathcal{G}]\!]) = \dot{\alpha}^\otimes(\mathsf{S}^1[\![\mathcal{G}]\!])$ where the *nonterminal productivity abstraction* is defined pointwise on terminal languages derived for nonterminals $\dot{\alpha}^\otimes \triangleq \boldsymbol{\lambda}\, L \bullet \boldsymbol{\lambda}\, A \bullet \alpha^\otimes(L(A))$ as true if the nonterminal can produce a non-empty terminal language and false otherwise $\alpha^\otimes \triangleq \boldsymbol{\lambda}\, \Sigma \bullet \left(\Sigma \neq \varnothing \;?\; \mathrm{tt} \;:\; \mathrm{ff}\right)$ so that $\langle \mathscr{N} \mapsto \wp(\mathscr{T}^\star),\ \dot{\subseteq} \rangle \underset{\dot{\alpha}^\otimes}{\overset{\dot{\gamma}^\otimes}{\leftrightarrows}} \langle \mathscr{N} \mapsto \mathbb{B},\ \dot{\Longrightarrow} \rangle$.The productivity iterative fixpoint computation [8, **Ex. 8.2.12**] is $\mathsf{S}^\otimes[\![\mathcal{G}]\!] = \mathbf{lfp}^{\dot{\Longrightarrow}}\, \hat{\dot{\mathsf{F}}}^\otimes[\![\mathcal{G}]\!]$ where the bottom-up transformer $\hat{\dot{\mathsf{F}}}^\otimes[\![\mathcal{G}]\!]$ is (14) instantiated as given in **Sect. 13**.

17 Top-down Grammar Analysis

17.1 Follow

The classical definition of FOLLOW [8, Sect. 8.2.8] provides information on the possible right context of nonterminals during syntax analysis. The *follow abstraction* $\alpha^f \in \mathscr{V}^\star \mapsto (\mathscr{N} \mapsto \wp(\mathscr{T} \cup \{\dashv\}))$ is

$$\begin{aligned}\alpha^f(\eta) \triangleq \boldsymbol{\lambda}\, A \bullet \{a \in \mathscr{T} \mid \exists \eta', \eta'' : \eta = \eta' A \eta'' \wedge \exists \eta''' \in \mathscr{T}^\star : \eta'' \overset{\star}{\Longmapsto}_{\mathcal{G}} a\eta'''\} \cup \\ \{\dashv \mid \exists \eta', \eta'' : \eta = \eta' A \eta'' \wedge \eta'' \overset{\star}{\Longmapsto}_{\mathcal{G}} \epsilon\}\end{aligned}$$

where we use the classical convention that sentences derived from the grammar axiom $\overline{S}$ are assumed to be followed by the extra symbol $\dashv \notin \mathscr{V}$ ($\dashv$ is # in [8, Sect. 8.2.8]). This is extended to $\alpha^f(\Sigma) \in \wp(\mathscr{V}^\star) \mapsto (\mathscr{N} \mapsto \wp(\mathscr{T} \cup \{\dashv\}))$ as $\alpha^f(\Sigma) \triangleq \boldsymbol{\lambda}\, A \bullet \bigcup_{\eta \in \Sigma} \alpha^f(\eta) A$ so that $\langle \wp(\mathscr{V}^\star),\ \subseteq \rangle \underset{\alpha^f}{\overset{\gamma^f}{\leftrightarrows}} \langle \mathscr{N} \mapsto \wp(\mathscr{T} \cup \{\dashv\}),\ \dot{\subseteq} \rangle$.The definition of FOLLOW [8, Def. 8.2.22] can also use that of FIRST since

Theorem 32 $\alpha^f(\Sigma) = \boldsymbol{\lambda}\, A \bullet \bigcup_{\eta' A \eta'' \in \Sigma} \hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!](\eta'')[\epsilon/\dashv]$ *where* $X[a/b] \triangleq (X \setminus \{a\}) \cup \{b \mid a \in X\}$. □

The *follow semantics* $\mathsf{S}^f[\![\mathcal{G}]\!]$ of a grammar $\mathcal{G}$ is $\mathsf{S}^f[\![\mathcal{G}]\!] \triangleq \alpha^f(\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!](\overline{S}))$ so that we get [8, Def. 8.2.22]

Theorem 33 $\mathsf{S}^f[\![\mathcal{G}]\!] = \boldsymbol{\lambda}\, A \bullet \{a \in \mathscr{T} \mid \exists \eta, \eta' : \overline{S} \overset{\star}{\Longmapsto}_{\mathcal{G}} \eta A a \eta'\} \cup \{\dashv \mid \exists \eta : \overline{S} \overset{\star}{\Longmapsto}_{\mathcal{G}} \eta A\}$.

By abstraction of the fixpoint characterization **Th. 27** of $\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!]$, we get the classical FOLLOW algorithm as an iterative fixpoint computation [8, **Fig. 8.13**]

Theorem 34 $\mathsf{S}^f[\![\mathcal{G}]\!] \subseteqq \mathit{lfp}^{\dot{\subseteqq}} \check{\mathsf{F}}^f[\![\mathcal{G}]\!]$ *where*

$$\check{\mathsf{F}}^f[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\, \phi \bullet \boldsymbol{\lambda}\, A \bullet \{\dashv \mid A = \overline{S}\} \cup \bigcup_{B \to \sigma A \sigma' \in \mathscr{R}} (\hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!](\sigma') \setminus \{\epsilon\}) \cup \left(\epsilon \in \hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!](\sigma') \;?\; \phi(B) : \varnothing\right) .$$

and $\subseteqq$ *denotes* $=$ *if all nonterminals in* $\mathcal{G}$ *are productive (as defined in* ***Sect. 16.3****) else* $\subseteqq$ *denotes* $\subseteq$. □

17.2 Nonterminal Accessibility

The classical definition of *accessible nonterminals* [8, Def. 8.2.4] provides information on which nonterminals of the grammar are used in the definition of the language generated for the grammar axiom. The *accessibility abstraction* is $\alpha^a \triangleq \boldsymbol{\lambda}\, \Sigma \bullet \boldsymbol{\lambda}\, A \bullet \left(\exists \sigma, \sigma' \in \mathscr{V}^\star : \sigma A \sigma' \in \Sigma \;?\; \mathrm{tt} : \mathrm{ff}\right)$ so that $\langle \mathscr{N} \mapsto \wp(\mathscr{V}^\star), \dot{\subseteq} \rangle \underset{\alpha^a}{\overset{\gamma^a}{\leftrightarrows}} \langle \mathscr{N} \mapsto \mathbb{B}, \dot{\Longrightarrow} \rangle$. The *nonterminal accessibility semantics* is $\mathsf{S}^a[\![\mathcal{G}]\!] \triangleq \alpha^a(\mathsf{S}^{\check{L}}[\![\mathcal{G}]\!](\overline{S}))$. This is the classical definition [8, Def. 8.2.4] since

Theorem 35 $\mathsf{S}^a[\![\mathcal{G}]\!] = \boldsymbol{\lambda}\, A \bullet \exists \sigma, \sigma' \in \mathscr{V}^\star : \overline{S} \overset{\star}{\Longmapsto}_{\mathcal{G}} \sigma A \sigma'$. □

The accessibility semantics $\mathsf{S}^a[\![\mathcal{G}]\!]$ has the following fixpoint characterization

Theorem 36 $\mathsf{S}^a[\![\mathcal{G}]\!] = \mathit{lfp}^{\dot{\subseteq}} \check{\mathsf{F}}^a[\![\mathcal{G}]\!]$ *where* $\check{\mathsf{F}}^a[\![\mathcal{G}]\!]\phi A \triangleq (A = \overline{S}) \vee \bigvee_{B \to \sigma A \sigma' \in \mathscr{R}} \phi(B)$. □

The accessibility semantics is an abstraction of the follow semantics since, if all nonterminals are productive (as defined in **Sect. 16.3**), a nonterminal is accessible if and only if it has a non-empty follow set.

Theorem 37 *(All nonterminals are productive)* $\Longrightarrow \left(\mathsf{S}^a[\![\mathcal{G}]\!] = \alpha^\otimes(\mathsf{S}^f[\![\mathcal{G}]\!])\right)$. □

18 Grammar Problem

Knuth's *grammar problem* [1], a generalization of the single-source shortest-path problem, is to compute the minimum-cost derivation of a terminal string from each non-terminal of a given *superior grammar* that is a context-free grammar, with rules of the form $A \to g(A_1, \ldots, A_n), n \geqslant 0$ (where 'g', '(', ',', and ')' are terminals), equipped with a cost function val such that the cost of a derivation is $\mathrm{val}(A \to g(A_1, \ldots, A_n)) = \mathrm{val}(g)(\mathrm{val}(A_1), \ldots, \mathrm{val}(A_n))$ and $\mathrm{val}(g) \in \mathbb{R}_+^n \mapsto \mathbb{R}_+$, $\mathbb{R}_+ \triangleq \{x \in \mathbb{R} \mid x \geqslant 0\} \cup \{\infty\}$, is a so-called *superior* function [1], a condition weakened in [2] where Knuth's algorithm is also given an incremental version.

Knuth's grammar problem [1] can be generalized to any bottom-up abstract grammar semantics $\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!]$ by considering $\alpha(\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!])$ where $\langle \hat{\mathsf{D}}^{\hat{\sharp}}, \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathbb{R}_+, \geqslant \rangle$ is a Galois connection and $\langle \mathbb{R}_+, \geqslant, \infty, 0, \min, \max \rangle$ is a complete lattice.

Knuth considers the particular case when $\mathsf{S}^{\hat{\sharp}}[\![\mathcal{G}]\!] = \mathsf{S}^{\ell}[\![\mathcal{G}]\!]$ and $\langle \hat{\mathsf{D}}^{\hat{\sharp}}, \sqsubseteq \rangle = \langle \wp(S), \subseteq \rangle$ where S is a set (indeed $S = \wp(\mathscr{T}^\star)$ in [1,2]) with $\alpha(X) \triangleq \min\{\mathrm{val}(x) \mid x \in X\}$ and $\gamma(m) \triangleq \{x \in S \mid \mathrm{val}(x) \geqslant m\}$. Since α is antitone, the corresponding abstract semantics is taken in terms of greatest fixpoints for $\leqslant$ [2]. Knuth's monotony hypothesis [1,2] ensures the existence of the greatest fixpoint. The rule soundness condition (15) then amounts to Knuth's hypothesis that for every nonterminal A, every string in $\mathsf{S}^{\ell}[\![\mathcal{G}]\!]A$ is a composition of superior functions $\alpha(g(x_1, \ldots, x_n)) = \mathrm{val}(g)(\alpha(x_1), \ldots, \alpha(x_n))$.

Knuth superiority condition [1] and its variant [2] ensure that the greatest fixpoint can be computed by an elimination algorithm (generalizing Dijkstra's algorithm to solve shortest path problems [17]). However in general one must resort to an infinite fixpoint iteration as shown with the choice of $S = \wp(\mathscr{T}^\star)$, $\mathrm{val}(x) = \frac{1}{|x|}$ so that $\mathrm{val}(g)() = \frac{1}{3}$ and $\mathrm{val}(g)(x_1, \ldots, x_n) = \frac{1}{\frac{1}{x_1} + \ldots + \frac{1}{x_n} + n + 2}$ which, for the grammar $A \to a()$, $A \to b(A, A)$ requires an infinite iteration and a passage to the limit 0.

Our generalization also copes with implicit abstractions of a grammar considered by [1,2] where a grammar is "recoded" into a superior grammar, which can indeed be defined by an appropriate α.

19 Bottom-Up Parsing

Given a grammar $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ and an input $\sigma = \sigma_1 \sigma_2 \ldots \sigma_n \in \mathscr{T}^\star$, $n \geq 0$, *parsing* consists in proving either $\sigma \in \mathsf{S}^{\ell}[\![\mathcal{G}]\!](\overline{S})$ or $\sigma \notin \mathsf{S}^{\ell}[\![\mathcal{G}]\!](\overline{S})$, that is, by **Th. 28**, providing an algorithmic answer to the question $\overline{S} \overset{\star}{\Longmapsto}_{\mathcal{G}} \sigma$?

Bottom-up parsing is an abstraction of a bottom-up grammar semantics by restriction to a given input sentence. This is illustrated with the Cocke-Younger-Kasami or CYK algorithm [4, Sect. 4.2.1] attributed by [18] to John Cocke, [19,20]). It is traditionally restricted to grammars $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ in *Chomsky normal form* with rules of the form $A \to BC$ and $A \to a$ where $A, B, C \in \mathscr{N}$ and $a \in \mathscr{T}$. We now design CYK by calculus for arbitrary grammars.

CYK is an abstract interpretation of the terminal language semantics $\mathsf{S}^{\ell}[\![\mathcal{G}]\!]$ by

$$\alpha^{CYK} \triangleq \boldsymbol{\lambda}\,\sigma \bullet \boldsymbol{\lambda}\,X \bullet \{\langle i,\, j\rangle \in \hat{\mathsf{D}}^{CYK}(\sigma) \mid \sigma_i \ldots \sigma_{i+j-1} \in X\} \tag{26}$$

where

$$\hat{\mathsf{D}}^{CYK} \triangleq \boldsymbol{\lambda}\,\sigma \bullet \{\langle i,\, j\rangle \mid i \in [1, |\sigma| + 1] \wedge j \in [0, |\sigma|] \wedge i + j \leq |\sigma| + 1\}$$

so that $\langle i,\, j\rangle$ denotes the subsentence of length j from position i in σ (in particular $\langle |\sigma| + 1,\, 0\rangle$ denotes the empty sentence ϵ after $\sigma = \sigma\epsilon$). Given $\sigma \in \mathscr{T}^\star$, we have

$$\langle \wp(\mathscr{T}^\star),\, \subseteq\rangle \underset{\alpha^{CYK}(\sigma)}{\overset{\gamma^{CYK}(\sigma)}{\leftrightarrows}} \langle \wp(\hat{\mathsf{D}}^{CYK}(\sigma)),\, \subseteq\rangle\ .$$

The pointwise extension to $\mathscr{N}$ is

$$\alpha^{CYK} \triangleq \boldsymbol{\lambda}\,\sigma \bullet \boldsymbol{\lambda}\,X \bullet \boldsymbol{\lambda}\,A \bullet \alpha^{CYK}(X(A)) \tag{27}$$

so that

$$\langle \mathscr{N} \mapsto \wp(\mathscr{T}^\star),\, \dot{\subseteq}\rangle \underset{\alpha^{CYK}(\sigma)}{\overset{\gamma^{CYK}(\sigma)}{\leftrightarrows}} \langle \mathscr{N} \mapsto \wp(\hat{\mathsf{D}}^{CYK}(\sigma)),\, \dot{\subseteq}\rangle\ .$$

The correctness of this parsing approach is proved by the following

Theorem 38 $\quad \sigma \in \mathsf{S}^\ell[\![\mathcal{G}]\!](\overline{S}) \Longleftrightarrow \langle 1,\, |\sigma|\rangle \in \alpha^{CYK}(\sigma)(\mathsf{S}^\ell[\![\mathcal{G}]\!])(\overline{S})$. □

The CYK algorithm is derived by abstracting the fixpoint definition **Th. 23** of $\mathsf{S}^\ell[\![\mathcal{G}]\!] = \mathbf{lfp}^{\dot{\subseteq}} \hat{\mathsf{F}}^{\rightarrow\ell}[\![\mathcal{G}]\!]$ by α^{CYK}.

Theorem 39

$$\alpha^{CYK}(\sigma)(\mathsf{S}^\ell[\![\mathcal{G}]\!])(\overline{S}) = \boldsymbol{lfp}^{\dot{\subseteq}} \hat{\mathsf{F}}^{\rightarrow CYK}[\![\mathcal{G}]\!](\sigma)$$

where

$$\hat{\mathsf{F}}^{\rightarrow CYK}[\![\mathcal{G}]\!] \in \wp(\hat{\mathsf{D}}^{CYK}) \mapsto \wp(\hat{\mathsf{D}}^{CYK})$$

$$\hat{\mathsf{F}}^{\rightarrow CYK}[\![\mathcal{G}]\!] \triangleq \boldsymbol{\lambda}\,\rho \bullet \boldsymbol{\lambda}\,A \bullet \bigcup_{A \to \sigma \in \mathscr{R}} \hat{\mathsf{F}}^{\rightarrow CYK}[A \to \boldsymbol{.}\sigma]\rho$$

$$\hat{\mathsf{F}}^{\rightarrow CYK}[A \to \sigma\boldsymbol{.}a\sigma'] \triangleq \boldsymbol{\lambda}\,\rho \bullet \{\langle i,\, j\rangle \in \hat{\mathsf{D}}^{CYK}(\sigma) \mid \sigma_i = a \wedge \langle i + 1,\, j - 1\rangle \in \hat{\mathsf{F}}^{\rightarrow CYK}[A \to \sigma a\boldsymbol{.}\sigma']\rho\}$$

$$\hat{\mathsf{F}}^{\rightarrow CYK}[A \to \sigma\boldsymbol{.}B\sigma'] \triangleq \boldsymbol{\lambda}\,\rho \bullet \{\langle i,\, j\rangle \in \hat{\mathsf{D}}^{CYK}(\sigma) \mid \exists k : 0 \leqslant k \leqslant j : \langle i,\, k\rangle \in \rho(B) \wedge \langle i + k,\, j - k\rangle \in \hat{\mathsf{F}}^{\rightarrow CYK}[A \to \sigma B\boldsymbol{.}\sigma']\rho\}$$

$$\hat{\mathsf{F}}^{\rightarrow CYK}[A \to \sigma\boldsymbol{.}] \triangleq \boldsymbol{\lambda}\,\rho \bullet \{\langle i,\, 0\rangle \mid 1 \leqslant i \leqslant |\sigma|\}$$ □

Because the abstract domain $\langle \mathscr{N} \mapsto \wp(\hat{\mathsf{D}}^{CYK}(\sigma)),\, \dot{\subseteq}\rangle$ is finite, the iterative computation of $\mathbf{lfp}^{\dot{\subseteq}} \mathsf{F}^{CYK}[\![\mathcal{G}]\!](\sigma)$ terminates whence by **Th. 39** and **Th. 38** so does the CYK parsing algorithm. The CYK dynamic programming algorithm organizes the computation of the pairs $\langle i,\, j\rangle \in \hat{\mathsf{D}}^{CYK}(\sigma)$ in order to avoid repetition of work already done.

20 Top-down Parsing

20.1 Nonrecursive Predictive Parser

A nonrecursive predictive parser is formally derived from the prefix derivation semantics $\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!]$ of **Sect. 5** by applying the abstraction

$$\alpha^{LL} \triangleq \boldsymbol{\lambda}\,\overline{S}\,\boldsymbol{\cdot}\,\boldsymbol{\lambda}\,\sigma\,\boldsymbol{\cdot}\,\boldsymbol{\lambda}\,X\,\boldsymbol{\cdot}\,\{\langle i,\,\varpi\rangle \mid \exists \theta = \varpi_0 \xrightarrow{\ell_0} \varpi_1 \ldots \varpi_{m-1} \xrightarrow{\ell_{m-1}} \varpi_m \in X.\overline{S} : i \in [0,|\sigma|] \wedge \alpha^{\tau}(\theta) = \sigma_1 \ldots \sigma_i \wedge \varpi = \varpi_m\}$$

where the *terminal abstraction* $\alpha^{\tau} \in \Theta \mapsto \mathscr{T}^{\star}$ collects terminal labels of derivations, as follows

$$\begin{array}{ll}
\alpha^{\tau}(\theta_1 \xrightarrow{(\!|A} \theta_2) \triangleq \alpha^{\tau}(\theta_1)\alpha^{\tau}(\theta_2) & \alpha^{\tau}(\varpi) \triangleq \epsilon, \quad \varpi \in \mathcal{S} \\
\alpha^{\tau}(\theta_1 \xrightarrow{A|)} \theta_2) \triangleq \alpha^{\tau}(\theta_1)\alpha^{\tau}(\theta_2) & \alpha^{\tau}(\vdash) \triangleq \epsilon \\
\alpha^{\tau}(\theta_1 \xrightarrow{a} \theta_2) \triangleq \alpha^{\tau}(\theta_1)a\alpha^{\tau}(\theta_2), \quad a \in \mathscr{T} & \alpha^{\tau}(\dashv) \triangleq \epsilon\ .
\end{array}$$

Let us write $\wp_1(S) \triangleq \{\{x\} \mid x \in S\}$ for the set of singletons of a set S and let $\alpha^{\bullet} \in \wp_1(S) \mapsto S$ be $\alpha^{\bullet}(\{x\}) \triangleq x$. We have

Lemma 40 $\forall \theta \in \Theta_{(\!|} : \alpha^{\tau}(\theta) = \alpha^{\bullet} \circ \alpha^{\ell} \circ \alpha^{\hat{L}} \circ \alpha^{\hat{s}} \circ \alpha^{\hat{\delta}}(\theta)$.

The interpretation of the pair $\langle i,\,\varpi_i\rangle$ is that in the left-to-right scanning of the input sentence σ up to position i, the prefix $\sigma_1 \ldots \sigma_i$ (ϵ when $i = 0$) has been recognized by a prefix derivation from the start symbol $\overline{S}$. The stack ϖ_i allows for the recognition of the rest of the sentence, if possible. Fixing the start symbol $\overline{S}$ and the input sentence σ, we have a Galois connection

$$\langle \wp(\Theta),\,\subseteq\rangle \underset{\alpha^{LL}(\overline{S})(\sigma)}{\overset{\gamma^{LL}(\overline{S})(\sigma)}{\leftrightarrows}} \langle \wp([0,|\sigma|] \times \mathcal{S}),\,\subseteq\rangle$$

The correctness of this parsing approach is proved by the following

Theorem 41 $\sigma \in \mathsf{S}^{\ell}[\![\mathcal{G}]\!](\overline{S}) \iff \langle |\sigma|,\,\dashv\rangle \in \alpha^{LL}(\overline{S})(\sigma)(\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!])$. □

To get a correct parsing algorithm, it remains

- to express $\alpha^{LL}(\overline{S})(\sigma)(\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!])$ in fixpoint form by abstraction of the fixpoint definition **Th. 6** of $\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!]$ (as shown in **Th. 42**), and
- to prove the termination of the fixpoint iteration (as shown in **Th. 44** for non left-recursive grammars).

Theorem 42 $\alpha^{LL}(\overline{S})(\sigma)(\mathsf{S}^{\vec{\partial}}[\![\mathcal{G}]\!]) = \mathit{lfp}^{\subseteq}\,\mathsf{F}^{LL}[\![\mathcal{G}]\!](\sigma)$ *where*

$$\begin{array}{l}
\mathsf{F}^{LL}[\![\mathcal{G}]\!](\sigma) \in \wp([0,|\sigma|] \times \mathcal{S}) \mapsto \wp([0,|\sigma|] \times \mathcal{S}) \\
\mathsf{F}^{LL}[\![\mathcal{G}]\!](\sigma) = \boldsymbol{\lambda}\,X\,\boldsymbol{\cdot}\,\{\langle 0,\vdash\rangle\} \cup \{\langle 0,\,\dashv[\overline{S} \to \boldsymbol{.}\eta]\rangle \mid \langle 0,\vdash\rangle \in X \wedge \overline{S} \to \eta \in \mathscr{R}\} \cup \\
\qquad \{\langle i+1,\,\varpi[A \to \eta a\boldsymbol{.}\eta']\rangle \mid \langle i,\,\varpi[A \to \eta\boldsymbol{.}a\eta']\rangle \in X \wedge a = \sigma_{i+1}\} \cup \\
\qquad \{\langle i,\,\varpi[A \to \eta B\boldsymbol{.}\eta'][B \to \boldsymbol{.}\varsigma]\rangle \mid \langle i,\,\varpi[A \to \eta\boldsymbol{.}B\eta']\rangle \in X \wedge B \to \varsigma \in \mathscr{R}\} \\
\qquad \cup \{\langle i,\,\varpi\rangle \mid \langle i,\,\varpi[A \to \eta\boldsymbol{.}]\rangle \in X\}\ .
\end{array}$$

□

$\mathbf{lfp}^{\subseteq} \mathsf{F}^{LL}[\![\mathcal{G}]\!](\sigma)$ is exactly the set of reachable states of the transition system $\langle [0, |\sigma|] \times \mathcal{S},\ \xrightarrow{\text{LL}} \rangle$ where

$$\langle 0,\ \vdash \rangle \xrightarrow{\text{LL}} \langle 0,\ \dashv [\overline{S} \to \centerdot \eta] \rangle \qquad \overline{S} \to \eta \in \mathscr{R} \tag{28}$$

$$\langle i,\ \varpi[A \to \eta \centerdot \sigma_{i+1} \eta'] \rangle \xrightarrow{\text{LL}} \langle i+1,\ \varpi[A \to \eta \sigma_{i+1} \centerdot \eta'] \rangle \tag{29}$$

$$\langle i,\ \varpi[A \to \eta \centerdot B \eta'] \rangle \xrightarrow{\text{LL}} \langle i,\ \varpi[A \to \eta B \centerdot \eta'][B \to \centerdot \varsigma] \rangle \qquad B \to \varsigma \in \mathscr{R} \tag{30}$$

$$\langle i,\ \varpi[A \to \eta \centerdot] \rangle \xrightarrow{\text{LL}} \langle i,\ \varpi \rangle \tag{31}$$

with initial state $\langle 0, \vdash \rangle$. By **Th. 41**, parsing is therefore reduced to proving that the final state $\langle |\sigma|, \dashv \rangle$ is reachable (which can be done by computing the iterates of $\mathsf{F}^{LL}[\![\mathcal{G}]\!](\sigma)$ or equivalently by exploring the descendants of the transition relation $\xrightarrow{\text{LL}}$ with backtracking when reaching a dead-end [4, Alg. 4.1, Sect. 4.1.3]).

Example 43 Consider the grammar $\mathcal{G} = \langle \{a, b\},\ \{A\},\ A,\ \{A \to A, A \to a\} \rangle$. For the input sentence $\sigma = a$ we have

$\langle 0, \vdash \rangle$ ⟅initial state⟆
$\xrightarrow{\text{LL}}$ $\langle 0, \dashv [A \to \centerdot a] \rangle$ ⟅by (28) with rule $A \to a$⟆
$\xrightarrow{\text{LL}}$ $\langle 1, \dashv [A \to a \centerdot] \rangle$ ⟅by (29) since $\sigma_1 = a$⟆
$\xrightarrow{\text{LL}}$ $\langle 1, \dashv \rangle$ ⟅by (31), which is a final state .⟆

On the other hand, the transitions for $\sigma = b$ either lead to dead ends or do not terminate

$\langle 0, \vdash \rangle$ ⟅initial state⟆
$\xrightarrow{\text{LL}}$ $\langle 0, \dashv [A \to \centerdot A] \rangle$ ⟅by (28) with rule $A \to A$ since $A \to a$ would lead to a dead end because $\sigma_1 = b \neq a$⟆
$\xrightarrow{\text{LL}}$ $\langle 0, \dashv [A \to \centerdot A][A \to \centerdot A] \rangle$ ⟅by (30) with rule $A \to A$ since $A \to a$ would lead to a dead end because $\sigma_1 = b \neq a$⟆
$\xrightarrow{\text{LL}}$ $\langle 0, \dashv [A \to \centerdot A][A \to \centerdot A][A \to \centerdot A] \rangle$ ⟅by (30) with rule $A \to A$ since $A \to a$ would lead to a dead end because $\sigma_1 = b \neq a$⟆
$\xrightarrow{\text{LL}}$ ... ⟅etc, ad infinitum, without any possibility of success or failure in a blocking state.⟆ □

Theorem 44 *The nonrecursive predictive parsing algorithm for a grammar* $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ *terminates (i.e. the transition relation* $\xrightarrow{\text{LL}}$ *has no infinite trace for all input sentences* $\sigma \in \mathscr{T}^\star$*) if and only if the grammar* $\mathcal{G}$ *has no* left recursion *(that is* $\exists A \in \mathscr{N} : \exists \eta \in \mathscr{V}^\star : A \overset{+}{\Longrightarrow}_{\mathcal{G}} A\eta$*).* □

20.2 Nonrecursive Predictive Parsing with Lookahead

The nondeterminism in predictive parsing can be reduced by driving the right context in derivations (as approximated using First and Follow). We start by elucidating the rôle of the right context in derivations.

Given a stack $\varpi = \dashv[A_1 \to \eta_1 \centerdot \eta_1'] \dots [A_p \to \eta_p \centerdot \eta_p']$, $p \geqslant 0$ where $\varpi = \dashv$ when $p = 0$, we define the *right context* $\varpi^{\triangle}$ *of* ϖ as

$$\varpi^{\triangle} \triangleq \eta_p' \eta_{p-1}' \dots \eta_2' \eta_1'$$

with $\eta_p' \eta_{p-1}' \dots \eta_2' \eta_1' = \epsilon$ when $p = 0$.

Theorem 45 *Let* $\varpi_0 \xrightarrow{\ell_0} \varpi_1 \dots \varpi_{i-1} \xrightarrow{\ell_{i-1}} \varpi_i \xrightarrow{\ell_i} \varpi_{i+1} \dots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n \in \mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!]$ *be a maximal derivation of the grammar* $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ *with* $i > 0$. *Then*

$$\varpi_i{}^{\triangle} \stackrel{\star}{\Longmapsto}_{\mathcal{G}} \alpha^{\tau}(\varpi_i \xrightarrow{\ell_i} \varpi_{i+1} \dots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n) \qquad \square$$

We call $\alpha^{\tau}(\varpi_i \xrightarrow{\ell_i} \varpi_{i+1} \dots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n)$ the *terminal right context* of ϖ_i.

In order to approximate the right contexts in derivations by their first symbol, we define

$$\begin{aligned}
&\hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!][A \to \eta \centerdot \eta'] \qquad (32)\\
&\triangleq \hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!](\eta') \oplus^1 \mathsf{S}^f[\![\mathcal{G}]\!](A)\\
&= \left(\mathsf{S}^f[\![\mathcal{G}]\!](A) \neq \varnothing \mathbin{?} (\hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!](\eta') \setminus \{\epsilon\}) \cup \left(\epsilon \in \hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!](\eta') \mathbin{?} \mathsf{S}^f[\![\mathcal{G}]\!](A) \mathbin{:} \varnothing\right) \mathbin{:} \varnothing\right)\\
&= \left(\mathsf{S}^f[\![\mathcal{G}]\!](A) \neq \varnothing \mathbin{?} (\hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!](\eta') \setminus \{\epsilon\}) \cup \left(\hat{\overrightarrow{\mathsf{S}^{\epsilon}}}[\![\mathcal{G}]\!](\eta') \mathbin{?} \mathsf{S}^f[\![\mathcal{G}]\!](A) \mathbin{:} \varnothing\right) \mathbin{:} \varnothing\right).
\end{aligned}$$

Corollary 46 *Let* $\varpi_0 \xrightarrow{\ell_0} \varpi_1 \dots \varpi_{i-1} \xrightarrow{\ell_{i-1}} \varpi_i \xrightarrow{\ell_i} \varpi_{i+1} \dots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n \in \mathsf{S}^{\hat{d}}[\![\mathcal{G}]\!].\overline{S}$, $i > 0$ *be a maximal derivation of the grammar* $\mathcal{G} = \langle \mathscr{T}, \mathscr{N}, \overline{S}, \mathscr{R} \rangle$ *from the grammar start symbol* $\overline{S}$. *Then*

$$\alpha^{\tau}(\varpi_i \xrightarrow{\ell_i} \varpi_{i+1} \dots \varpi_{n-1} \xrightarrow{\ell_{n-1}} \varpi_n)\dashv = a\sigma$$

where $\varpi_i = \varpi_i'[A \to \eta \centerdot \eta']$, $a \in \mathscr{T} \cup \{\dashv\}$, $\sigma \in (\mathscr{T} \cup \{\dashv\})^{\star}$ *and*

$$a \in \hat{\overrightarrow{\mathsf{S}^1}}[\![\mathcal{G}]\!][A \to \eta \centerdot \eta'] \,. \qquad \square$$

If the input sentence σ derives from the start symbol $\overline{S}$ then the right context $\varpi^{\triangle}$ of the stack ϖ in $\langle i, \varpi \rangle$ should derive in the rest $\sigma_{i+1} \dots \sigma_n$ of the input sentence. In order to introduce a lookahead, this can be approximated by the fact that, according to **Cor. 46**, the first symbol of this right context should be σ_{i+1} (which, by definition, is $\dashv$ when $i = n$ so that $\sigma_{|\sigma|+1} \triangleq \dashv$).

$$\alpha^{LL(1)} \triangleq \boldsymbol{\lambda}\,\overline{S}\bullet\boldsymbol{\lambda}\,\sigma\bullet\boldsymbol{\lambda}\,X\bullet\{\langle i,\,\varpi\rangle \mid \exists\theta=\varpi_0\xrightarrow{\ell_0}\varpi_1\ldots\varpi_{m-1}\xrightarrow{\ell_{m-1}}\varpi_m\in X.\overline{S}:\\ i\in[0,|\sigma|]\wedge\alpha^{\tau}(\theta)=\sigma_1\ldots\sigma_i\wedge\varpi=\varpi_m\wedge\forall\varpi'\in\mathcal{S},A\to\eta\eta'\in\mathscr{R}:\\ (\varpi=\varpi'[A\to\eta.\eta']\wedge i\leqslant|\sigma|)\Longrightarrow(\sigma_{i+1}\in\overrightarrow{\hat{\mathsf{S}}^{1}}[\![\mathcal{G}]\!][A\to\eta.\eta'])\}\ .$$

The correctness of the nonrecursive predictive parser with lookahead is established by the following

Theorem 47

$$\sigma\in\mathsf{S}^{\ell}[\![\mathcal{G}]\!](\overline{S})\Longleftrightarrow\langle|\sigma|,\,\dashv\rangle\in\alpha^{LL(1)}(\overline{S})(\sigma)(\overrightarrow{\mathsf{S}^{\partial}}[\![\mathcal{G}]\!])\ .$$ □

The nonrecursive predictive parser with lookahead is obtained by expressing the abstract semantics in fixpoint form

Theorem 48

$$\alpha^{LL(1)}(\overline{S})(\sigma)(\overrightarrow{\mathsf{S}^{\partial}}[\![\mathcal{G}]\!])=\mathit{lfp}^{\subseteq}\,\mathsf{F}^{LL(1)}[\![\mathcal{G}]\!](\sigma)$$

where $\mathsf{F}^{LL(1)}[\![\mathcal{G}]\!](\sigma)\in\wp([0,|\sigma|]\times\mathcal{S})\mapsto\wp([0,|\sigma|]\times\mathcal{S})$ *is*

$$\begin{aligned}\mathsf{F}^{LL(1)}[\![\mathcal{G}]\!](\sigma)=\boldsymbol{\lambda}\,X\bullet\{\langle 0,\,\vdash\rangle\}\cup\\ \{\langle 0,\,\dashv[\overline{S}\to.\eta]\rangle\mid\langle 0,\,\vdash\rangle\in X\wedge\overline{S}\to\eta\in\mathscr{R}\wedge\\ \sigma_1\in\overrightarrow{\hat{\mathsf{S}}^{1}}[\![\mathcal{G}]\!][\overline{S}\to.\eta]\}\cup\\ \{\langle i+1,\,\varpi[A\to\eta a.\eta']\rangle\mid\langle i,\,\varpi[A\to\eta.a\eta']\rangle\in X\wedge\\ a=\sigma_{i+1}\wedge\sigma_{i+2}\in\overrightarrow{\hat{\mathsf{S}}^{1}}[\![\mathcal{G}]\!][A\to\eta a.\eta']\}\cup\\ \{\langle i,\,\varpi[A\to\eta B.\eta'][B\to.\varsigma]\rangle\mid\langle i,\,\varpi[A\to\eta.B\eta']\rangle\in X\wedge\\ B\to\varsigma\in\mathscr{R}\wedge\sigma_{i+1}\in\overrightarrow{\hat{\mathsf{S}}^{1}}[\![\mathcal{G}]\!][B\to.\varsigma]\}\cup\\ \{\langle i,\,\varpi\rangle\mid\langle i,\,\varpi[A\to\eta.]\rangle\in X\}\ .\end{aligned}\tag{33}$$ □

Again, observe that $\mathbf{lfp}^{\subseteq}\,\mathsf{F}^{LL(1)}[\![\mathcal{G}]\!](\sigma)$ is exactly the set of reachable states of the transition system $\langle[0,|\sigma|]\times\mathcal{S},\,\overset{\text{LL(1)}}{\longrightarrow}\rangle$ where

$$\langle 0,\,\vdash\rangle\overset{\text{LL(1)}}{\longrightarrow}\langle 0,\,\dashv[\overline{S}\to.\eta]\rangle\qquad\overline{S}\to\eta\in\mathscr{R}\wedge\ \sigma_1\in\overrightarrow{\hat{\mathsf{S}}^{1}}[\![\mathcal{G}]\!][\overline{S}\to.\eta]\tag{34}$$

$$\langle i,\,\varpi[A\to\eta.\sigma_{i+1}\eta']\rangle\overset{\text{LL(1)}}{\longrightarrow}\langle i+1,\,\varpi[A\to\eta\sigma_{i+1}.\eta']\rangle\qquad\sigma_{i+2}\in\overrightarrow{\hat{\mathsf{S}}^{1}}[\![\mathcal{G}]\!][A\to\eta a.\eta']\tag{35}$$

$$\langle i,\,\varpi[A\to\eta.B\eta']\rangle\overset{\text{LL(1)}}{\longrightarrow}\langle i,\,\varpi[A\to\eta B.\eta'][B\to.\varsigma]\rangle\ B\to\varsigma\in\mathscr{R}\wedge\ \sigma_{i+1}\in\overrightarrow{\hat{\mathsf{S}}^{1}}[\![\mathcal{G}]\!][B\to.\varsigma]\}\tag{36}$$

$$\langle i,\,\varpi[A\to\eta.]\rangle\overset{\text{LL(1)}}{\longrightarrow}\langle i,\,\varpi\rangle\tag{37}$$

with initial state $\langle 0,\,\vdash\rangle$. This is essentially the algorithm suggested at the end of [4, Sect. 4.1.4] to speed up top-down nondeterministic parsing.

Indeed the lookahead may been done freely between the two extremes of everywhere in **Th. 47** and nowhere **Th. 41**, as follows

Corollary 49 *If* $\mathsf{F}^{LL(1)}[\![\mathcal{G}]\!](\sigma) \subseteq \mathsf{F}[\![\mathcal{G}]\!](\sigma) \subseteq \mathsf{F}^{LL}[\![\mathcal{G}]\!](\sigma)$ *then*

$$\sigma \in \mathsf{S}^{\ell}[\![\mathcal{G}]\!](\overline{S}) \iff \langle|\sigma|,\ \dashv\rangle \in \mathit{lfp}^{\subseteq}\ \mathsf{F}[\![\mathcal{G}]\!](\sigma)\ .$$

The iterative computation of $\mathit{lfp}^{\subseteq}\ \mathsf{F}[\![\mathcal{G}]\!](\sigma)$ *terminates for all* σ *if and only if the grammar* $\mathcal{G}$ *has no left recursion.* □

Our presentation of LL(1) parsing differs from the classical introduction in [8], mainly because, for practical efficiency and simplicity reasons, only the table-driven deterministic case is classically considered.

21 Conclusion

Many meanings assigned to grammars (such as syntax tree, protolanguage or terminal language generation) and grammar manipulation algorithms (such as grammar flow analyses or parsers) have quite similar structures. We have shown that this is because they are all abstract interpretations of a grammar small-step operational semantics to derive sentences together with their structure.

Future work should include the extension of the approach to context-free grammars such as *contextual grammars* [21] or to mildly context-sensitive grammars attempting to express the formal power needed to define the syntax of natural languages by tree rewriting such as (multicomponent) tree adjoining grammars or, more generally, *range concatenation grammars* [22].

Acknowledgements We thank Tom Reps for drawing our attention to [1,2].

References

1. Knuth, D.: A generalization of Dijkstra's algorithm. Inf. Process. Lett. **6**(1) (Feb. 1977) 1–5
2. Ramalingam, G., Reps, T.: An incremental algorithm for a generalization of the shortest-path problem. J. Algorithms **21**(2) (1996) 267–305
3. Bar-Hillel, J., Perles, M., Shamir, E.: On formal properties of simple phrase structure grammars. Z. Phonetik. Sprachwiss. Kommunikationforsch. **14** (1961) 143–172
4. Aho, A., Ullman, J.: Parsing. Volume 1 of The Theory of Parsing, Translation and Compiling. Prentice-Hall (1972)
5. Möncke, U., Wilhelm, R.: Iterative algorithms on grammar graphs. In Schneider, H., Gottler, H., eds.: Proc. 8^{th} Conf. on Graphtheoretic Concepts in Computer Science (WG'82), Hanser Verlag (1982) 177–194
6. Möncke, U.: Generierung von Systemen zur Transformation attributierter Operatorbäume; Komponenten des Systems und Mechanismen der Generierung. Diplomarbeit, Universität des Saarlandes, Saarbrücken (1985)
7. Möncke, U., Wilhelm, R.: Grammar flow analysis. In Alblas, H., Melichar, B., eds.: Attribute Grammars, Applications and Systems, Intl. Summer School SAGA, Prague, CZ, 4–13 June , 1991, Proc. Volume 545 of LNCS., Springer (1991) 151–186

8. Wilhelm, R., Maurer, D.: Übersetzerbau. Theorie, Konstruktion, Generierung. Springer (1992)
9. Cousot, P., Cousot, R.: Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: 4^{th} POPL, Los Angeles, CA, ACM Press (1977) 238–252
10. Chomsky, N.: Three models for the description of language. IEEE Trans. Information Theory **2**(3) (1956) 113–124
11. Chomsky, N.: Syntactic Structures. Mouton, de Gruyter (1957)
12. Cousot, P.: Constructive design of a hierarchy of semantics of a transition system by abstract interpretation. Theoret. Comput. Sci. **277**(1—2) (2002) 47–103
13. Cousot, P., Cousot, R.: Systematic design of program analysis frameworks. In: 6^{th} POPL, San Antonio, TX, ACM Press (1979) 269–282
14. Cousot, P., Cousot, R.: Constructive versions of Tarski's fixed point theorems. Pacific J. Math. **82**(1) (1979) 43–57
15. Ginsburg, S., Rice, G.: Two families of languages related to ALGOL. J. ACM **9** (1962) 350–371
16. Schützenberger, M.: On a theorem of R. Jungen. Proc. Amer. Math. Soc. **13** (1962) 885–889
17. Dijkstra, E.: A note on two problems in connexion with graphs. Numer. Math. **1** (1959) 269–271
18. Hays, D.: Introduction to Computational Linguistics. Amer. Elsevier (1967)
19. Younger, D.: Recognition and parsing of context-free languages in time n^3. Inform. and Control **10**(2) (1967) 609–617
20. Kasami, T.: An efficient recognition and syntax analysis algorithm for context-free languages. Technical report, Air Force Cambridge Research Laboratory, Bedford, MA, US (Aug. 1965)
21. Ehrenfeucht, A., Päun, G., Rozenberg, G.: Contextual grammars and formal languages. In Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages. Volume 2. Springer (1997) 237–293
22. Boullier, P.: From contextual grammars to range concatenation grammars. ENTCS **53** (Apr. 2001) 41–52 http://www.elsevier.nl/locate/entcs/volume53.html.