Dessins Mots $\rightarrow$ Phrases Sens Bin./Exéc.

Analyse syntaxique

Grammaires : classification

- Toutes
- Si $w \to w' \in R$ alors $|w| \le |w'|$ $\alpha^n\beta^n\gamma^n$
- Hors contexte : $w \to w' \in R$ alors $w \in V_n$
- Linéaires... (et donc régulières)

$$\alpha^n\beta^n\gamma^n \leadsto \begin{cases} S & \to & \alpha ST\gamma \mid \alpha T\gamma \\ \gamma T & \to & T\gamma \\ T & \to & \beta \end{cases}$$

Analyse syntaxique : dérivations g/d

Jusqu'ici : symboles $\rightarrow$ $\boxed{\text{transducteur}}$ $\rightarrow$ tokens

Expressions arith. sur $\{+; \times; (;); id; cte\}$

$$E \to id \mid cte \mid E+E \mid E \times E \mid (E) \qquad \leadsto \qquad x + 2.5 \times 4 + (y+z)$$

$$E \to \underline{E} + E \to id + \underline{E} \to id + \underline{E} \times E \to id + cte \times \underline{E} \to id + cte \times \underline{E} + E \to$$
$$id + cte \times cte + \underline{E} \to id + cte \times cte + (\underline{E}) \to id + cte \times cte + (\underline{E} + E) \to$$
$$id + cte \times cte + (id + \underline{E}) \to id + cte \times cte + (id + id)$$

Dérivation gauche

Parenthésage implicite : $(x + (2.5 \times (4 + (y+z))))$

Analyse syntaxique : dérivations g/d

Jusqu'ici : symboles $\rightarrow$ $\boxed{\text{transducteur}}$ $\rightarrow$ tokens

Expressions arith. sur $\{+; \times; (;); id; cte\}$

$$E \to id \mid cte \mid E+E \mid E \times E \mid (E) \qquad \leadsto \qquad x + 2.5 \times 4 + (y+z)$$

$$E \to E + \underline{E} \to E + (\underline{E}) \to E + (E + \underline{E}) \to E + (\underline{E} + id) \to \underline{E} + (id + id) \to$$
$$E \times \underline{E} + (id + id) \to \underline{E} \times cte + (id+id) \to E + \underline{E} \times cte + (id + id) \to$$
$$\underline{E} + cte \times cte + (id + id) \to id + cte \times cte + (id + id)$$

Dérivation droite

Parenthésage implicite : $(((x + 2.5) \times 4) + (y+z))$

Analyse syntaxique : dérivations g/d

$(((x+2.5)\times 4)+(y+z)) \neq (x+(2.5\times(4+(y+z))))$ Ambigu

Dérivation droite avec $((x+(2.5\times 4))+(y+z))$?

G ambiguë si $\exists w \in \mathcal{L}(G)$ tel que plusieurs dérivations droite pour w.

Vers non ambiguë ?

Bad news : pas toujours possible. . .

$\rightsquigarrow$ et pour cas faciles ?

- Nouveau NON terminal par niveau priorité
- Récursif gauche si associatif gauche (resp. droite)

Analyse syntaxique : dérivations g/d

$$E \rightarrow id \mid cte \mid E+E \mid E\times E \mid (E) \qquad \rightsquigarrow \qquad x+2.5\times 4+(y+z)$$

$+ < \times < \cdots$

$+, \times$ assoc. gauche

$$\begin{array}{lcl} E & \rightarrow & E+F \mid F \\ F & \rightarrow & F\times G \mid G \\ G & \rightarrow & cte \mid id \mid (E) \end{array}$$

Unique dérivation gauche ou droite (exo)

Un peu plus complexe, un peu plus long mais univoque

Analyse syntaxique : arbres syntaxiques

Plusieurs dérivations pour un même résultat (permutations, etc.)

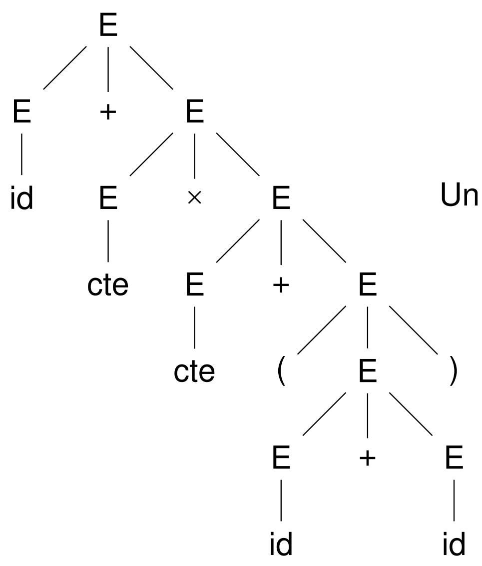
$\rightsquigarrow$ Représentation invariante Repr. unique lorsque G non ambiguë

Soit $G = (V_t, v_n, S, R)$, arbres de syntaxe de G :

- Nœuds internes étiquetés par V_n
- Feuilles étiquetées par V
- Si nœud interne N a pour k fils $\alpha_1, \ldots, \alpha_k$ alors $N \to \alpha_1 \ldots \alpha_k \in R$

Arbre de dérivation : $\Lambda = S$ et feuilles $\in V_t$

Analyse syntaxique : arbres syntaxiques



Un arbre dans grammaire ambiguë

Analyse syntaxique : arbres syntaxiques

Arbre de dérivation = plusieurs dérivations

$\rightsquigarrow$ Stratégies de parcours (parent avant fils)

Pour arbre de dérivation : mot des feuilles $\in \mathcal{L}(G)$

Réciproque ? : production arbre, récurrence sur long. dérivation

- Nulle : arbre = feuille α
- $N \xrightarrow{n} w_1 M w_2 \rightarrow w_1 \alpha_1 \cdots \alpha_k w_2$:

  ajout de k fils au nœud M de l'arbre de $N \xrightarrow{n} w_1 M w_2$

Analyse syntaxique : arbres syntaxiques

Arbre de dérivation = plusieurs dérivations

Théorème.

Pour G, $w \in V_t^\star$ dans $\mathcal{L}(G)$ ssi arbre de dérivation pour w dans G.

G ambiguë si $\exists w \in \mathcal{L}(G)$ avec deux arbres de dérivations distincts.

Construction de l'arbre ? vers le haut ou vers le bas ?

Analyse syntaxique : descendante

Exemple : grammaire de Dick

$$S \rightarrow \varepsilon \mid (S)S$$

$((()())()) \in \mathcal{L}$?

Facile :

- Lecture à partir de la gauche — Left
- Construction dérivation gauche — Left
- Règle déterminée par 1 caractère à produire — 1

$\rightsquigarrow$ LL(1)

Exo : arbre

Analyse syntaxique : descendante

- Efficace
- Simple

- Pas toutes LL(1) !
- Si récursive gauche : échec (Raté pour arith.)

Analyse syntaxique : ascendante

$$\begin{array}{lcl} E & \to & E+F \mid F \\ F & \to & F\times G \mid G \\ G & \to & cte \mid id \mid (E) \end{array}$$

Récursive gauche. . .

$E \to F$ ou $E \to E+F$?

$\leadsto$ Analyse ascendante

- Lecture à partir de gauche
- Dérivation droite — à l'envers

En particulier construction d'une forêt

Analyse syntaxique : ascendante

$$\begin{array}{lcl} E & \to & E+F \mid F \\ F & \to & F\times G \mid G \\ G & \to & cte \mid id \mid (E) \end{array}$$

En particulier construction d'une forêt

Racine de forêt : suite des racines des arbres la constituant

Opérations :

- Lecture — Shift
- Enracinement — Reduce

  Sur juxtaposition ff' construction de fN si $N \to \Lambda(f')$

Analyse syntaxique : ascendante

$E \to E+F \mid F \qquad F \to F\times G \mid G \qquad G \to cte \mid id \mid (E)$

- Lecture
- Enracinement : sur ff' construction de fN si $N \to \Lambda(f')$

$id \times id + id$?

Ex. Séq. d'id

Conflit Shift/Reduce $\leadsto$ convention Shift

Conflit Reduce/Reduce $\leadsto$ modifier la grammaire

Syntaxe abstraite

Type et signification : depuis l'arbre de syntaxe

Utilisateur : ce qu'on écrit

Concrète : presque comme on écrit — Impropre à bonne compréhension

$\leadsto$ niveau intermédiaire : syntaxe abstraite — dépolluée

On veut :

- Sans ambiguïté
- Sans scories
- Structure et valeurs

Syntaxe abstraite : grammaire

T ensemble d'étiquettes abstraites

$V_t = T \cup \{(;);,\}$

G sur V_t terminal abstraite si

- Règles : $N \to t$ ou $N \to t(N_1, \ldots, N_k)$ pour $N_i \in V_n, t \in T$
- Occurrence t unique dans G

Mot généré : expression abstraite

$T = \{p, m, cte\}$ $E \to p(E, E) \mid m(E, E) \mid cte$ g. abstraite pour arith.

Syntaxe abstraite : grammaire

T ensemble d'étiquettes abstraites

$V_t = T \cup \{(;);,\}$

G sur V_t terminal abstraite si

- Règles : $N \to t$ ou $N \to t(N_1, \ldots, N_k)$ pour $N_i \in V_n, t \in T$
- Occurrence t unique dans G

Mot généré : expression abstraite

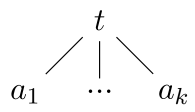
$T = \{+, \times, cte\}$ $E \to +(E, E) \mid \times(E, E) \mid cte$ g. abstraite pour arith.

Syntaxe abstraite

Objectif 1 :

G abstraite nécessairement non ambiguë OK

Représentation arborescente : facile grâce à la forme des règles

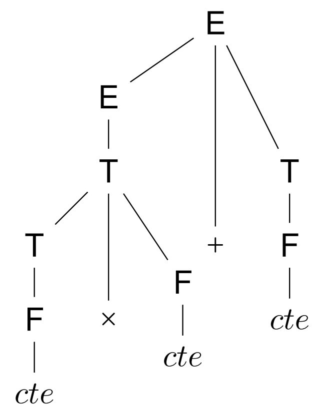


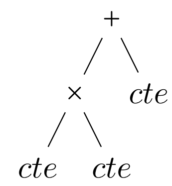
Arbre de syntaxe abstraite A de sorte N : nœuds dans T et

- $N \to t \in R$ et $A = t$ ou bien
- $N \to t(N_1, \ldots, N_k) \in R$ et $\Lambda(A) = t$
  A a alors k fils ASA de sortes respectives $N_1 \ldots N_k$

Syntaxe abstraite

Comparaison AS / ASA





Objectif 2 : OK

Syntaxe abstraite : dans ASA

Pour valeurs : étiquette = langage régulier

$$\begin{array}{lcl} E & \to & +(E,E) \mid \times(E,E) \mid cte(Nat) \\ Nat & \to & (\mathtt{0}|\mathtt{1}|\ldots|\mathtt{9})+ \end{array}$$

Objectif 3 : OK

Syntaxe abstraite : construction ASA

Actions dans une grammaire

$$\begin{array}{lcll} E & \to & E+F & \{+(E_1,F_1)\} \\ E & \to & F & \{F_1\} \\ F & \to & F\times G & \{\times(G_1,F_1)\} \\ F & \to & G & \{G_1\} \\ G & \to & Nat & \{cte(\mathtt{val}(Nat_1))\} \\ G & \to & (E) & \{E_1\} \end{array}$$

Erreurs ici : SYNTAXIQUES