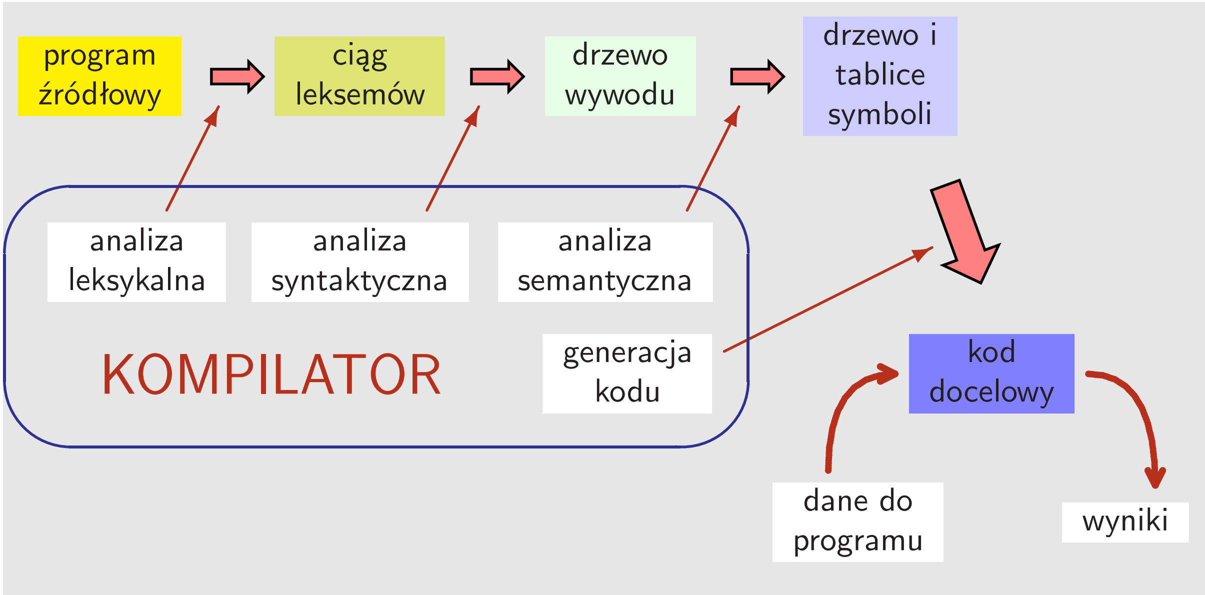
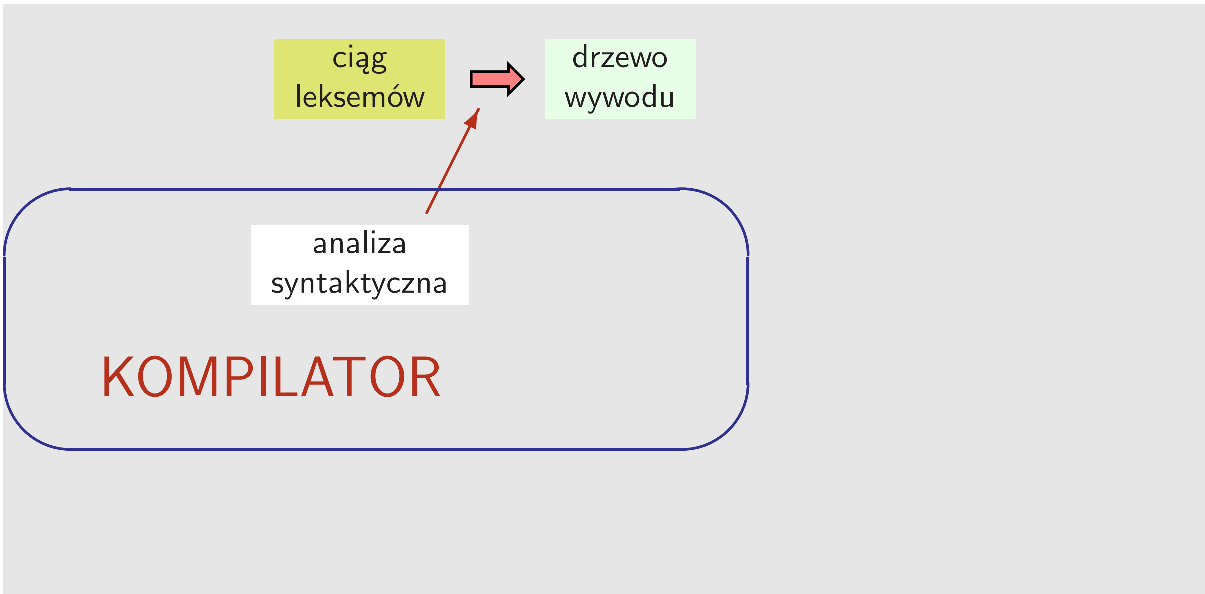
Uproszczony schemat kompilacji



Uproszczony schemat kompilacji



Budowanie drzewa wywodu

$$\left.\begin{array}{l}\text{analiza składniowa}\\ \text{analiza syntaktyczna}\\ \text{parsing}\\ \text{rozbiór}\end{array}\right\}\ \begin{array}{l}\text{synonimy oznaczające}\\ \text{budowę drzewa wywodu wg gramatyki}\end{array}$$

Budowy drzewa wywodu można dokonywać

- albo od strony korzenia (rozbiór *zstępujący* — *top-down*),
- albo od strony liści (rozbiór *wstępujący* — *bottom-up*).

Jest wiele metod rozbioru danego słowa; różnią się

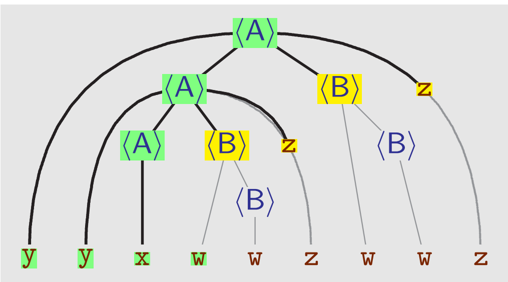
- zakresem stosowalności (do jakich rodzajów gramatyk bezkontekstowych się nadają),
- oraz efektywnością (jak szybko działają i ile potrzebują pamięci).

Budowanie drzewa wywodu

Przykład:

$$\begin{array}{lcl}\langle A\rangle & ::= & \mathtt{y}\langle A\rangle\langle B\rangle\mathtt{z} \mid \mathtt{x}\\ \langle B\rangle & ::= & \mathtt{w}\langle B\rangle \mid \mathtt{w}\end{array}$$

TOP-DOWN:

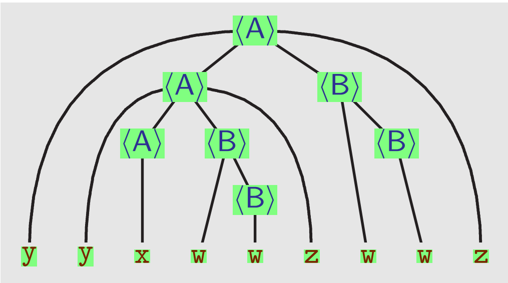


Budowanie drzewa wywodu

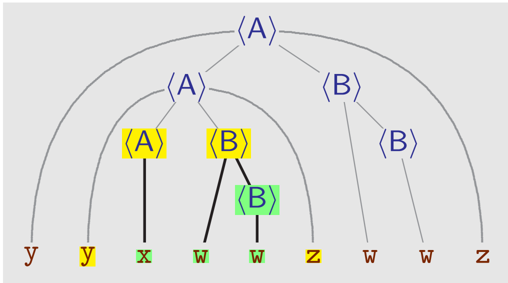
Przykład:

$$\langle A\rangle ::= \mathtt{y}\langle A\rangle\langle B\rangle\mathtt{z} \mid \mathtt{x}$$
$$\langle B\rangle ::= \mathtt{w}\langle B\rangle \mid \mathtt{w}$$

TOP-DOWN:



BOTTOM-UP:



- Metodą *top-down* drzewo buduje się od korzenia. Decyzje, jaką produkcję zastosować, podejmuje się w oparciu o opracowywany nieterminal oraz początek ciągu na wejściu.
- Metodą *bottom-up* drzewo buduje się od liści, zastępując fragmenty wejścia nieterminalami, z których do tych fragmentów prowadzi produkcja.

Koszty

- Do każdej **jednoznacznej** gramatyki bezkontekstowej można zbudować parser, działający w czasie $\mathcal{O}(n^3)$, gdzie n jest długością ciągu terminali na wejściu.
- Ta złożoność wynika z działania przez próbowanie i wycofywanie się w przypadku złego dopasowania; drzewo wywodu jest wielokrotnie budowane i niszczone.
- Niezbyt bolesne ograniczenia na gramatyki umożliwiają istnienie parsera, działającego w czasie $\mathcal{O}(n)$, czyli znacznie szybciej.
- **Wszystkie** praktycznie stosowane parsery działają w czasie $\mathcal{O}(n)$.

Rekursywny parser zstępujący

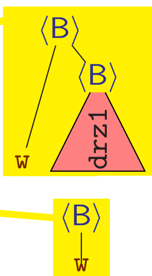
- Analiza *top-down*.
- Po jednej funkcji rekursywnej dla każdego nieterminala. Funkcja dla nieterminalu $\langle\mathsf{X}\rangle$
  - wywoływana jest po to, żeby skonstruować drzewo wywodu początkowych leksemów z wejścia, mające w korzeniu nieterminal $\langle\mathsf{X}\rangle$;
  - żeby móc zadecydować, której produkcji z nieterminalu $\langle\mathsf{X}\rangle$ użyć, jest wywoływana w sytuacji, w której znany (wczytany) jest już pierwszy liść (leksem) tego drzewa; kończy działanie w sytuacji, kiedy znany jest już pierwszy leksem *spoza* korony tego drzewa;
  - może zasygnalizować błąd i program ją wywołujący stara się znaleźć inną produkcję do zastosowania.

Rekursywny parser zstępujący

```
Boolean B(drzewo* drz) {
 drzewo  drz1;
 if (nowyleks('w'))
  if (B(&drz1)){
   *drz = ...;
   return true;
  }
  else {
   *drz = ...;
   return true;
  }
 else  return false;
}
```

```
Boolean  A(drzewo* drz);
Boolean  B(drzewo* drz);
```

$\langle\mathsf{B}\rangle ::= \mathtt{w}\langle\mathsf{B}\rangle \mid \mathtt{w}$

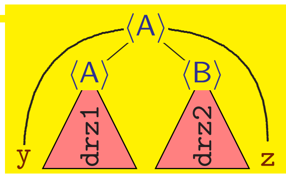


Rekursywny parser zstępujący

```
Boolean A(drzewo* drz) {
 drzewo  drz1, drz2;
 if (nowyleks('y'))
  if (A(&drz1))
   if (B(&drz2))
    if (nowyleks('z')) {
     *drz = ...;
     return true;
    }
    else blad("brakuje terminalu  z");
   else blad("brakuje nieterminalu  <B>");
  else blad("brakuje nieterminalu  <A>");
 else
  if (nowyleks('x')) {
   *drz = ...;
   return true;
  }
  else  return false;
}
```

```
Boolean
  A(drzewo* drz);
Boolean
  B(drzewo* drz);
```

$\langle \mathsf{A} \rangle ::= \mathtt{y}\langle \mathsf{A} \rangle\langle \mathsf{B} \rangle\mathtt{z} \mid \mathtt{x}$



Zakres stosowalności parserów zstępujących

DEFINICJA: Niech $\langle \Sigma, N, P, \langle \mathsf{S} \rangle \rangle$ będzie gramatyką bezkontekstową, niech $n \in \mathbb{N}$ będzie liczbą naturalną, niech $\alpha \in (\Sigma \cup N)^*$ będzie ciągiem terminali i nieterminali. Wtedy

$$\mathit{FIRST}_n(\alpha) \stackrel{\text{def}}{=} \left\{ \mathtt{a}_1\mathtt{a}_2 \ldots \mathtt{a}_n \in \Sigma^* \,\middle|\, \alpha \Rightarrow^* \mathtt{a}_1\mathtt{a}_2 \ldots \mathtt{a}_n\beta \text{ dla pewnego } \beta \in \Sigma^* \right\}$$

Czyli $\mathit{FIRST}_n(\alpha)$ jest zbiorem tych n-literowych napisów terminalowych, od których mogą się zaczynać napisy wyprowadzalne z α.

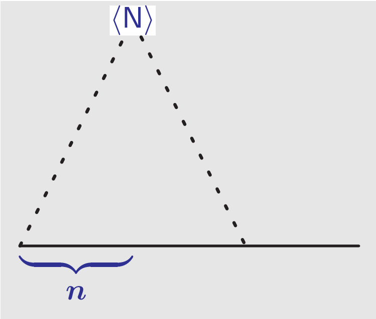
DEFINICJA: Gramatyka należy do klasy $\boldsymbol{LL(n)}$, jeżeli spełnia następujący warunek:

jeśli jakiś nieterminal ma więcej niż jedną produkcję

$$\langle \mathsf{N} \rangle ::= \alpha \mid \beta \mid \ldots$$

to $\mathit{FIRST}_n(\alpha) \cap \mathit{FIRST}_n(\beta) = \emptyset$.

Zakres stosowalności parserów zstępujących



To znaczy: jeśli

- wiemy nad którym nieterminalem ⟨N⟩ pracujemy, oraz
- znamy n pierwszych leksemów z ciągu, który ma stanowić koronę drzewa o korzeniu w ⟨N⟩,

to wiemy, którą produkcję zastosować.

Gramatyki dotąd prezentowane należały do $LL(1)$.

Zakres stosowalności parserów zstępujących

Przykład:

$$\begin{array}{rcl} \langle\text{miara}\rangle & ::= & \langle\text{odległość}\rangle \mid \langle\text{waga}\rangle \\ \langle\text{odległość}\rangle & ::= & \langle\text{cyfra}\rangle\mathtt{m} \mid \langle\text{cyfra}\rangle\langle\text{odległość}\rangle \\ \langle\text{waga}\rangle & ::= & \langle\text{cyfra}\rangle\mathtt{kg} \mid \langle\text{cyfra}\rangle\langle\text{waga}\rangle \end{array}$$

$$\langle\text{miara}\rangle \Rightarrow \langle\text{odległość}\rangle \Rightarrow^* \mathtt{11111m}$$

$$\langle\text{miara}\rangle \Rightarrow \langle\text{waga}\rangle \Rightarrow^* \mathtt{11111kg}$$

Który wywód zastosować?

Nie da się ustalić na podstawie pierwszej cyfry, bo nie wiadomo jeszcze, czy na końcu będzie `m` czy `kg`.

Parsery zstępujące nadają się tylko do gramatyk LL(n).

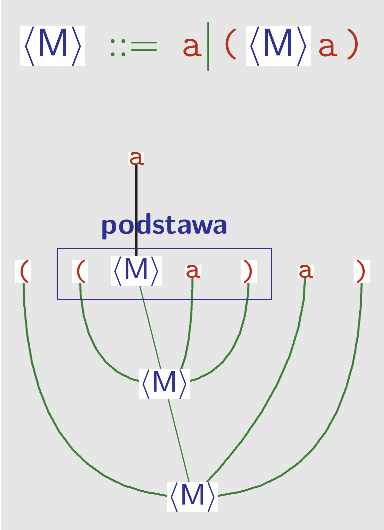
Rozbiór wstępujący — gramatyki z pierwszeństwem

Rozbiór wstępujący — budujemy drzewo rozbioru od liści.

W ciągu symboli wejściowych musimy znaleźć *podstawę* czyli *uchwyt* najbliższej redukcji, czyli podciąg który zostanie zredukowany w pierwszym kroku.

Następnie dokonujemy redukcji.

Znowu wyszukujemy *podstawę*.



Rozbiór wstępujący — gramatyki z pierwszeństwem

Rozbiór wstępujący — budujemy drzewo rozbioru od liści.

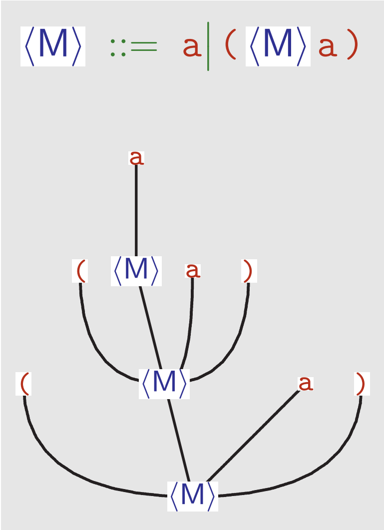
W ciągu symboli wejściowych musimy znaleźć *podstawę* czyli *uchwyt* najbliższej redukcji, czyli podciąg który zostanie zredukowany w pierwszym kroku.

Następnie dokonujemy redukcji.

Znowu wyszukujemy *podstawę*.

I dokonujemy redukcji.

Te kroki powtarzamy aż do skonstruowania całego drzewa.



W jaki sposób znajdujemy podstawę redukcji?

Rozbiór wstępujący — gramatyki z pierwszeństwem

W jaki sposób znajdujemy podstawę redukcji?

Relacje wpisane w tablicę związane są z kolejnością redukowania poszczególnych symboli:

- jeśli $x \doteq y$ to symbole x i y będą się redukować razem (jeśli stoją koło siebie i jeden należy do podstawy, to drugi też);
- jeśli $x \lessdot y$ to symbol x powinien zaczekać, aż y zostanie zredukowany (jeśli stoją koło siebie i y należy do podstawy, to x nie należy do podstawy);
- jeśli $x \gtrdot y$ to symbol y powinien zaczekać, aż x zostanie zredukowany (jeśli stoją koło siebie i x należy do podstawy, to y nie należy do podstawy);
- jeśli nie ma między nimi żadnej relacji, to nie powinny stać obok siebie (jeśli stoją koło siebie, to jest to błąd).

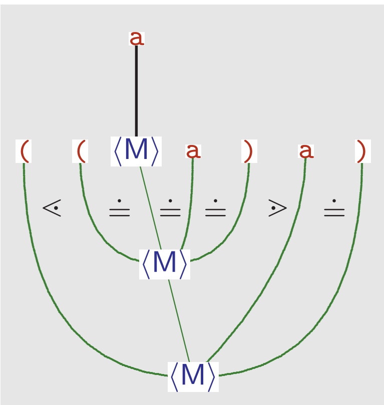
$\langle \mathrm{M} \rangle$ ::= a | ($\langle \mathrm{M} \rangle$ a)

Tablica pierwszeństwa redukcji:

	(	$\langle \mathrm{M} \rangle$	a	)
(	$\lessdot$	$\doteq$	$\lessdot$	
$\langle \mathrm{M} \rangle$			$\doteq$	
a			$\gtrdot$	$\doteq$
)			$\gtrdot$	

Rozbiór wstępujący — gramatyki z pierwszeństwem

W jaki sposób znajdujemy podstawę redukcji?



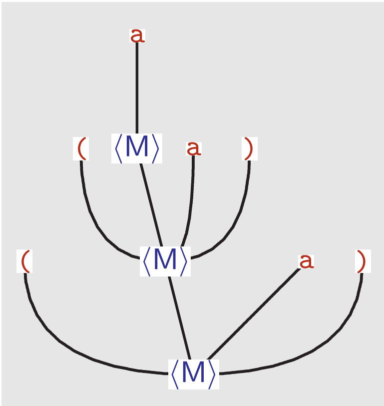
$\langle \mathrm{M} \rangle$ ::= a | ($\langle \mathrm{M} \rangle$ a)

Tablica pierwszeństwa redukcji:

	(	$\langle \mathrm{M} \rangle$	a	)
(	$\lessdot$	$\doteq$	$\lessdot$	
$\langle \mathrm{M} \rangle$			$\doteq$	
a			$\gtrdot$	$\doteq$
)			$\gtrdot$	

Rozbiór wstępujący — gramatyki z pierwszeństwem

W jaki sposób znajdujemy podstawę redukcji?



⟨M⟩ ::= a | (⟨M⟩ a)

Tablica pierwszeństwa redukcji:

	(	⟨M⟩	a	)
(	$\lessdot$	$\doteq$	$\lessdot$	
⟨M⟩			$\doteq$	
a			$\gtrdot$	$\doteq$
)			$\gtrdot$	

- **Jak skonstruować tablicę pierwszeństw?**
- **Dla jakich gramatyk da się to zrobić?**

Konstrukcja tablic pierwszeństwa

Niech $G = \langle \Sigma, N, P, S\rangle$ będzie gramatyką bezkontekstową.

DEFINICJA:

Niech $A \in N$.

$$\mathit{fst}\,A \stackrel{\text{def}}{=} \left\{x \in \Sigma \cup N \,\middle|\, \text{istnieje w } P \text{ jakaś produkcja } A \to \boldsymbol{x}v\right\}$$

$$\mathit{fst}^+\,A \stackrel{\text{def}}{=} \mathit{fst}\,A \cup \bigcup\left\{\mathit{fst}^+\,B \,\middle|\, B \in N \wedge B \in \mathit{fst}\,A\right\}$$

$$\mathit{lst}\,A \stackrel{\text{def}}{=} \left\{x \in \Sigma \cup N \,\middle|\, \text{istnieje w } P \text{ jakaś produkcja } A \to w\boldsymbol{x}\right\}$$

$$\mathit{lst}^+\,A \stackrel{\text{def}}{=} \mathit{lst}\,A \cup \bigcup\left\{\mathit{lst}^+\,B \,\middle|\, B \in N \wedge B \in \mathit{lst}\,A\right\}$$

Przykład:

⟨M⟩ ::= a | (⟨M⟩ a) — $\mathit{fst}\,\langle\mathrm{M}\rangle = \{\mathrm{a}, (\}$

Przykład:

$$\left.\begin{array}{lll} \langle\mathrm{M}\rangle & ::= & \mathrm{a} \mid \langle\mathrm{Q}\rangle\ \mathrm{a}\ ) \\ \langle\mathrm{Q}\rangle & ::= & (\ \langle\mathrm{M}\rangle \end{array}\right\} \quad — \quad \begin{cases} \mathit{fst}\,\langle\mathrm{M}\rangle = \{\mathrm{a}, \langle\mathrm{Q}\rangle\} \\ \mathit{fst}^+\,\langle\mathrm{M}\rangle = \{\mathrm{a}, \langle\mathrm{Q}\rangle\} \cup \mathit{fst}^+\,\langle\mathrm{Q}\rangle \\ \qquad\qquad = \{\mathrm{a}, \langle\mathrm{Q}\rangle, (\} \end{cases}$$

Konstrukcja tablic pierwszeństwa

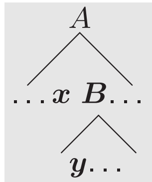
Niech $G = \langle \Sigma, N, P, S \rangle$ będzie gramatyką bezkontekstową.

DEFINICJA:

Relacje $\doteq$, $\lessdot$, $\gtrdot$ w $\Sigma \cup N$ określone są następująco:

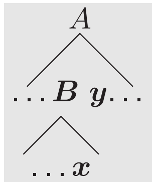
$x \doteq y \overset{\text{def}}{\Longleftrightarrow}$ w P istnieje produkcja $A \rightarrow w\boldsymbol{x}\boldsymbol{y}v$

$x \lessdot y \overset{\text{def}}{\Longleftrightarrow}$ w P istnieje produkcja $A \rightarrow w\boldsymbol{x}\boldsymbol{B}v$
taka że $y \in \mathit{fst}^+\, B$



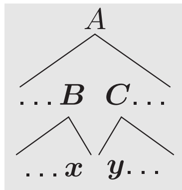
$x \gtrdot y \overset{\text{def}}{\Longleftrightarrow}$

w P istnieje produkcja $A \rightarrow w\boldsymbol{B}\boldsymbol{y}v$
taka że $x \in \mathit{lst}^+\, B$



lub

w P istnieje produkcja $A \rightarrow w\boldsymbol{B}\boldsymbol{C}v$
taka że $x \in \mathit{lst}^+\, B$ i $y \in \mathit{fst}^+\, C$



Konstrukcja tablic pierwszeństwa

Przykład:

$\langle \mathrm{M} \rangle$ ::= a | ($\langle \mathrm{M} \rangle$ a)

$x \doteq y \overset{\text{def}}{\Longleftrightarrow}$ istnieje produkcja $A \rightarrow w\boldsymbol{x}\boldsymbol{y}v$

($\doteq$ $\langle \mathrm{M} \rangle$ $\quad$ $\langle \mathrm{M} \rangle$ $\doteq$ a $\quad$ a $\doteq$)

$x \lessdot y \overset{\text{def}}{\Longleftrightarrow}$ istnieje produkcja $A \rightarrow w\boldsymbol{x}\boldsymbol{B}v$ taka że $y \in \mathit{fst}^+\, B$

($\lessdot$ a $\quad$ ($\lessdot$ ($\quad$ (bo $\mathit{fst}^+\, \langle \mathrm{M} \rangle = \{$a, ($\}$)

$x \gtrdot y \overset{\text{def}}{\Longleftrightarrow}$

istnieje produkcja $A \rightarrow w\boldsymbol{B}\boldsymbol{y}v$ taka że $x \in \mathit{lst}^+\, B$

a $\gtrdot$ a $\quad$) $\gtrdot$ a $\quad$ (bo $\mathit{lst}^+\, \langle \mathrm{M} \rangle = \{$a,)$\}$)

lub istnieje produkcja $A \rightarrow w\boldsymbol{B}\boldsymbol{C}v$ taka że $x \in \mathit{lst}^+\, B$ i $y \in \mathit{fst}^+\, C$

Konstrukcja tablic pierwszeństwa

	(	⟨M⟩	a	)
(	$\lessdot$	$\doteq$	$\lessdot$	
⟨M⟩			$\doteq$	
a			$\gtrdot$	$\doteq$
)			$\gtrdot$	

Uwaga:

- nie jest prawdą, że zawsze $x \doteq x$ (np. nie zachodzi a $\doteq$ a);
- nie jest prawdą, że zawsze z $x \doteq y$ wynika $y \doteq x$ (np. zachodzi a $\doteq$) ale nie zachodzi) $\doteq$ a);
- nie jest prawdą, że zawsze z $x \lessdot y$ wynika $y \gtrdot x$ (np. zachodzi ($\lessdot$ a ale nie zachodzi a $\gtrdot$ ( );
- nie jest prawdą, że zawsze z $x \lessdot y$ i $y \doteq z$ wynika $x \lessdot z$ (np. zachodzi ($\lessdot$ a i a $\doteq$) ale nie zachodzi ($\lessdot$)).

Zakres stosowalności gramatyk z pierwszeństwem

Przykład:

⟨W⟩ ::= ⟨S⟩ | ⟨W⟩ + ⟨S⟩
⟨S⟩ ::= ℓ | (⟨W⟩)

Ponieważ (i ⟨W⟩ sąsiadują w tej samej produkcji, więc ($\doteq$ ⟨W⟩.
Ponieważ ⟨W⟩ $\in$ *fst*$^+$ ⟨W⟩ , więc ($\lessdot$ ⟨W⟩.
Dla tej gramatyki nie da się więc skonstruować tablicy pierwszeństwa.

Przykład:

⟨A⟩ ::= a | ⟨B⟩ a
⟨B⟩ ::= a

Nie wiadomo, do jakiego nieterminalu redukować a. Dla tej gramatyki nie da się zastosować analizy z pierwszeństwem.

DEFINICJA:

Gramatyka bezkontekstowa $G = \langle \Sigma, N, P, S \rangle$ jest *gramatyką z pierwszeństwem* jeśli

- dla każdych dwóch symboli $x, y \in \Sigma \cup N$, zachodzi najwyżej jeden ze związków $x \doteq y$, $x \lessdot y$, $x \gtrdot y$ (może nie zachodzić żaden),
- nie istnieją dwie różne produkcje o tej samej prawej stronie, ani produkcja o pustej prawej stronie: ⟨A⟩ $\to \lambda$.

Poprawianie na gramatykę z pierwszeństwem

Przykład:

$\langle W\rangle ::= \langle S\rangle \mid \langle W\rangle + \langle S\rangle$
$\langle S\rangle ::= \ell \mid (\langle W\rangle)$

	$\langle W\rangle$	$\langle S\rangle$	+	ℓ	(	)
$\langle W\rangle$			$\doteq$			$\doteq$
$\langle S\rangle$			$\gtrdot$			$\gtrdot$
+		$\doteq$		$\lessdot$	$\lessdot$	
ℓ			$\gtrdot$			$\gtrdot$
(	$\doteq / \lessdot$	$\lessdot$		$\lessdot$	$\lessdot$	
)			$\gtrdot$			$\gtrdot$

$\langle W\rangle ::= \langle V\rangle$
$\langle V\rangle ::= \langle S\rangle \mid \langle V\rangle + \langle S\rangle$
$\langle S\rangle ::= \ell \mid (\langle W\rangle)$

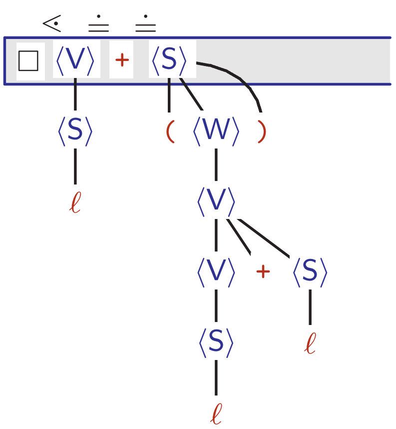
	$\langle W\rangle$	$\langle S\rangle$	+	ℓ	(	)	$\langle V\rangle$
$\langle W\rangle$						$\doteq$	
$\langle S\rangle$			$\gtrdot$			$\gtrdot$	
+		$\doteq$		$\lessdot$	$\lessdot$		
ℓ			$\gtrdot$			$\gtrdot$	
(	$\doteq$	$\lessdot$		$\lessdot$	$\lessdot$		$\lessdot$
)			$\gtrdot$			$\gtrdot$	
$\langle V\rangle$			$\doteq$			$\gtrdot$	

Użycie tablicy pierwszeństwa do rozbioru

$\langle W\rangle ::= \langle V\rangle$
$\langle V\rangle ::= \langle S\rangle \mid \langle V\rangle + \langle S\rangle$
$\langle S\rangle ::= \ell \mid (\langle W\rangle)$

	$\langle W\rangle$	$\langle S\rangle$	+	ℓ	(	)	$\langle V\rangle$	$\square$
$\langle W\rangle$						$\doteq$		$\gtrdot$
$\langle S\rangle$			$\gtrdot$			$\gtrdot$		$\gtrdot$
+		$\doteq$		$\lessdot$	$\lessdot$			$\gtrdot$
ℓ			$\gtrdot$			$\gtrdot$		$\gtrdot$
(	$\doteq$	$\lessdot$		$\lessdot$	$\lessdot$		$\lessdot$	$\gtrdot$
)			$\gtrdot$			$\gtrdot$		$\gtrdot$
$\langle V\rangle$			$\doteq$			$\gtrdot$		$\gtrdot$
$\square$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	

STOS:



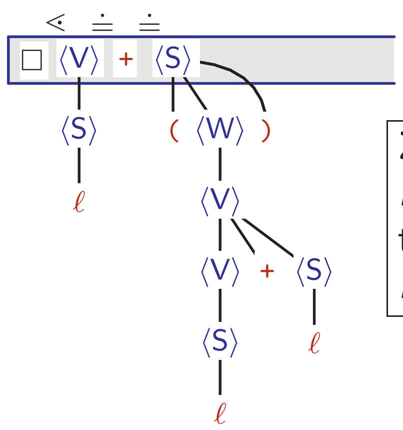
$\gtrdot$ WEJŚCIE: $\ell + (\ell + \ell) \,|\, + \ell \,\square$

Użycie tablicy pierwszeństwa do rozbioru

⟨W⟩ ::= ⟨V⟩
⟨V⟩ ::= ⟨S⟩ | ⟨V⟩ + ⟨S⟩
⟨S⟩ ::= ℓ | (⟨W⟩)

	⟨W⟩	⟨S⟩	+	ℓ	(	)	⟨V⟩	□
⟨W⟩						$\doteq$		$\gtrdot$
⟨S⟩			$\gtrdot$			$\gtrdot$		$\gtrdot$
+		$\doteq$		$\lessdot$	$\lessdot$			$\gtrdot$
ℓ			$\gtrdot$			$\gtrdot$		$\gtrdot$
(	$\doteq$	$\lessdot$		$\lessdot$	$\lessdot$		$\lessdot$	$\gtrdot$
)			$\gtrdot$			$\gtrdot$		$\gtrdot$
⟨V⟩			$\doteq$			$\gtrdot$		$\gtrdot$
□	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	

STOS:



$\gtrdot$ WEJŚCIE: ℓ + (ℓ + ℓ) | + ℓ □

Zakończenie pracy:
Na stosie: tylko □ i jedno drzewo z nieterminalem początkowym w korzeniu.
Na wejściu: tylko □.

Gramatyki z pierwszeństwem — podsumowanie

- Bezpośrednia *wstępująca* konstrukcja drzewa wywodu z użyciem stosu.
- **Zalety:**
  - prostota;
  - o wiele szybsze działanie i mniejsza zajętość pamięci niż w rozbiorach zstępujących;
  - możliwość dobrej sygnalizacji błędów.
- **Wady:**
  - konieczność poprawienia gramatyki przez dodanie nowych nieterminali i produkcji — czasem bardzo wielu;
  - dużo pracy.