Лекции по теории формальных языков

Лекция 13.
LR(0)-, SLR(1)-, LR(1)- и LALR(1)-анализ

Александр Сергеевич Герасимов
http://gas-teach.narod.ru

Кафедра математических и информационных технологий
Санкт-Петербургского академического университета
Российской академии наук.
Весенний семестр 2010/11 учебного года

13 мая 2011 г.

План

План

Анализ с помощью LR(0)-автомата: неформальное описание

- Подадим цепочку w на вход LR(0)-автомату $\mathcal{A}_G$ грамматики G.
- После каждого такта работы автомата $\mathcal{A}_G$ кладём новое состояние автомата в стек и, анализируя это состояние, до перехода к следующему такту производим описанные ниже действия.
- Если в состоянии есть пункт $[A \to \alpha_1 \cdot \alpha_2]$ и $\alpha_2 \neq \varepsilon$, то выполняем перенос следующего символа, т. е. переходим к следующему такту работы автомата $\mathcal{A}_G$.
- Если в состоянии есть пункт $[A \to \alpha\cdot]$, то выполняем свёртку по правилу $A \to \alpha$ (при этом активный префикс $\gamma\alpha$ в стеке заменяется на активный префикс γA):
  - выталкиваем из стека $|\alpha|$ состояний;
  - на вершине стека оказывается состояние `I` после прочтения γ;
  - вернём автомат в состояние `I` и дадим прочитать ему на следующем такте A.
- Если таким образом автомат прочитает w и придёт в состояние, содержащее пункт $[S' \to S\cdot]$, то $w \in L(G)$, иначе $w \notin L(G)$.

Конфликты

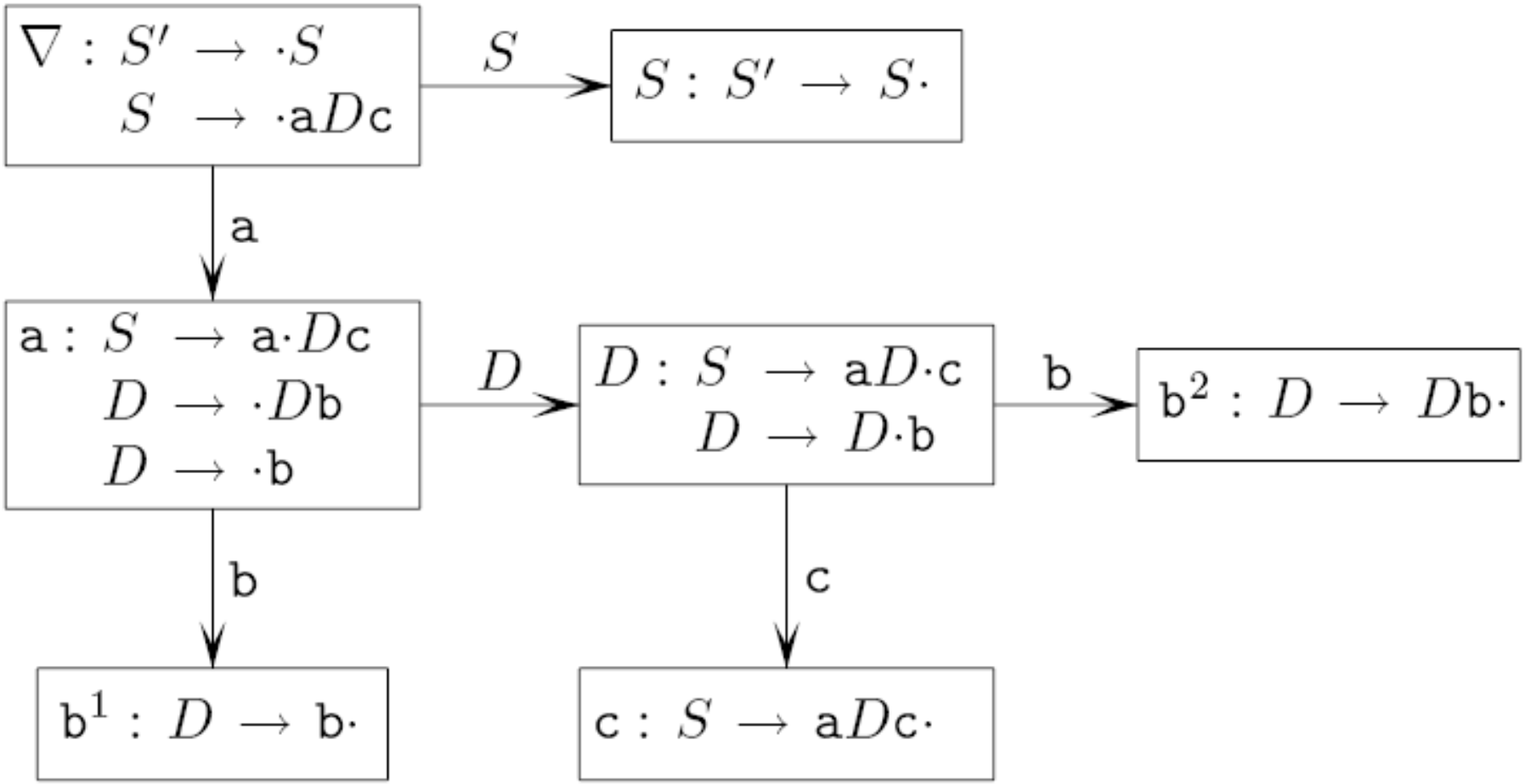
- *Конфликт перенос-свёртка*: в одном состоянии LR(0)-автомата имеются пункты
  - $[A \to \alpha_1 \cdot \alpha_2]$, где $\alpha_2 \neq \varepsilon$, и
  - $[B \to \beta\cdot]$.

- *Конфликт свёртка-свёртка*: в одном состоянии LR(0)-автомата имеются различные пункты
  - $[A \to \alpha\cdot]$ и
  - $[B \to \beta\cdot]$.

LR(0)-грамматика: определение и пример

Грамматика называется *LR(0)-грамматикой*, если каждое состояние её LR(0)-автомата, содержащее пункт вида $[A \to \alpha\cdot]$, состоит из единственного пункта.

Пример. G_1 : (0) $S' \to S$, (1) $S \to \mathtt{a}D\mathtt{c}$, (2) $D \to D\mathtt{b}$, (3) $D \to \mathtt{b}$.



Алгоритм построения LR(0)-анализатора

Вход. Расширенная LR(0)-грамматика $G = (\Sigma, \Gamma, P, S')$.
Выход. Таблица LR-анализа для грамматики G.

1. построить LR(0)-автомат $\mathcal{A}_G = (Q, \Sigma \cup \Gamma, \delta, \mathtt{I}_0, Q)$;
2. **для каждого** $\mathtt{I} \in Q$
3. **для каждого** $\mathtt{i} \in \mathtt{I}$
4. **если** $(\mathtt{i} = [S' \to S\cdot])$ $\textsc{action}(\mathtt{I}, \dashv) := \checkmark$;
5. **иначе если** $(\mathtt{i} = [A \to \alpha\cdot])$
6. **для каждого** $\mathtt{a} \in \Sigma \cup \{\dashv\}$
7. $\textsc{action}(\mathtt{I}, \mathtt{a}) := (\otimes\text{номер}(A \to \alpha))$;
8. **иначе если** $(\mathtt{i} = [A \to \beta_1 \cdot \mathtt{a}\beta_2])$
9. $\textsc{action}(\mathtt{I}, \mathtt{a}) := (\leftarrow \delta(\mathtt{I}, \mathtt{a}))$;
10. **иначе** (т. е. $\mathtt{i} = [A \to \beta_1 \cdot B\beta_2]$)
11. $\textsc{goto}(\mathtt{I}, B) := \delta(\mathtt{I}, B)$

Почему в каждой клетке построенной таблицы не более одной записи?

Пример построения LR(0)-анализатора

$G_1 : (0)\ S' \to S,\ (1)\ S \to \mathtt{a}D\mathtt{c},\ (2)\ D \to D\mathtt{b},\ (3)\ D \to \mathtt{b}.$

	ACTION				GOTO	
	a	b	c	$\dashv$	S	D
S				✓		
D		$\leftarrow \mathtt{b}^2$	$\leftarrow \mathtt{c}$			
a		$\leftarrow \mathtt{b}^1$				D
$\mathtt{b}^1$	$\otimes 3$	$\otimes 3$	$\otimes 3$	$\otimes 3$		
$\mathtt{b}^2$	$\otimes 2$	$\otimes 2$	$\otimes 2$	$\otimes 2$		
c	$\otimes 1$	$\otimes 1$	$\otimes 1$	$\otimes 1$		
∇	$\leftarrow \mathtt{a}$				S	

Предложение о корректности алгоритма построения LR(0)-анализатора

Предложение

LR(0)-анализатор, построенный алгоритмом на слайде 7 по грамматике G, допускает цепочку w тогда и только тогда, когда $w \in L(G)$.

Доказательство.

- Этот анализатор работает как LR(0)-автомат с запуском из неочередного состояния в случае свёртки и складыванием в состояний стек.
- Анализатор выполняет свёртку в том и только в том случае, когда он находится в состоянии `I` вида $\{[A \to \alpha\cdot]\}$.
- Если LR(0)-автомат находится в состоянии `I`, а в стеке — $\gamma = \gamma'\alpha$, то по основной теореме LR-анализа γ является активным префиксом, для которого допустим пункт $[A \to \alpha\cdot]$, а α является основой некоторой r-формы, начинающейся с γ.

Предложение о корректности алгоритма построения LR(0)-анализатора: окончание доказательства

- В состоянии I имеется единственный пункт $[A \to \alpha\cdot]$, поэтому α является основой любой r-формы, начинающейся с γ.
- Таким образом, анализатор выполняет свёртку тогда и только тогда, когда наверху стека находится основа текущей r-формы.
- Иначе анализатор пытается осуществить перенос. Если конкатенация содержимого стека и необработанного суффикса входной цепочки является r-формой, то такой перенос возможен, поскольку LR(0)-автомат распознаёт все активные префиксы.
- Наконец, допуск производится тогда и только тогда, когда входная цепочка свёрнута в аксиому грамматики.

Теорема об оценке числа шагов LR(0)-анализатора

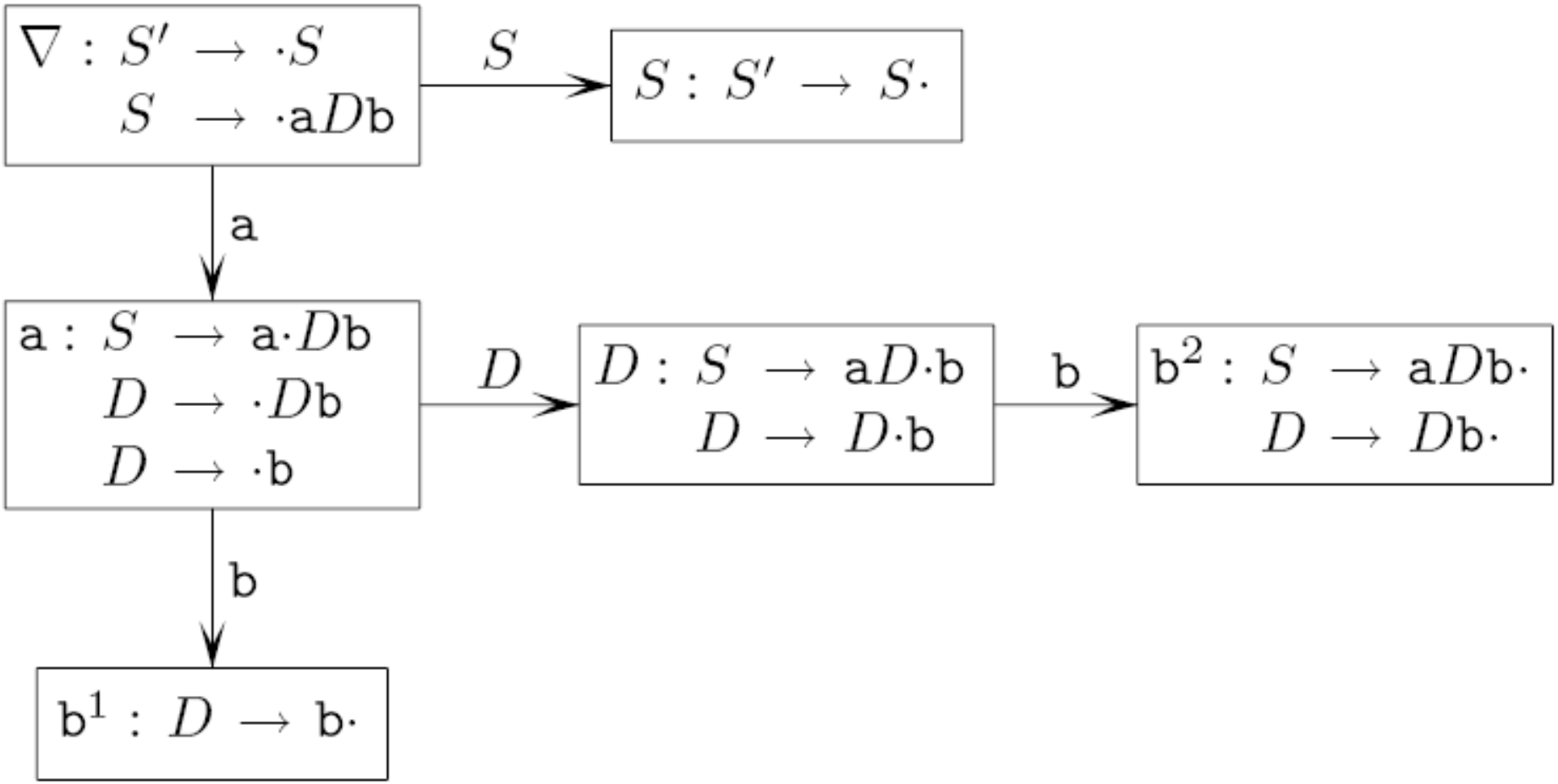
Теорема

Число шагов (переносов и свёрток) LR(0)-анализатора, построенного алгоритмом на слайде 7, линейно зависит от длины входной цепочки.

План

Пример конфликта «свёртка-свёртка»

G_2 : (0) $S' \to S$, (1) $S \to \mathtt{a}D\mathtt{b}$, (2) $D \to D\mathtt{b}$, (3) $D \to \mathtt{b}$.



Разрешение конфликта:

- если очередной символ $\dashv$, то $\otimes 1$;
- если очередной символ b, то $\otimes 2$.

Пример конфликта «перенос-свёртка»

$G_3 : (0)\ S' \to S,\ (1)\ S \to \mathtt{ac},\ (2)\ S \to \mathtt{a}D\mathtt{bc},\ (3)\ D \to D\mathtt{b},\ (4)\ D \to \varepsilon.$

$$\nabla = \{[S' \to \cdot S], [S \to \cdot\mathtt{ac}], [S \to \cdot\mathtt{a}D\mathtt{bc}]\} \xrightarrow{S} S = \{[S' \to S\cdot]\}$$
$$\downarrow \mathtt{a}$$
$$\mathtt{a} = \{[S \to \mathtt{a}\cdot\mathtt{c}], [S \to \mathtt{a}\cdot D\mathtt{bc}], [D \to \cdot D\mathtt{b}], [D \to \cdot]\} \xrightarrow{\mathtt{c}} \mathtt{c}^1 = \{[S \to \mathtt{ac}\cdot]\}$$
$$\downarrow D$$
$$D = \{[S \to \mathtt{a}D\cdot\mathtt{bc}], [D \to D\cdot\mathtt{b}]\} \xrightarrow{\mathtt{b}} \mathtt{b} = \{[S \to \mathtt{a}D\mathtt{b}\cdot\mathtt{c}], [D \to D\mathtt{b}\cdot]\} \xrightarrow{\mathtt{c}}$$
$$\mathtt{c}^2 = \{[S \to \mathtt{a}D\mathtt{bc}\cdot]\}$$

Разрешение конфликтов:

- если очередной символ c, то перенос;
- если очередной символ b, то свёртка.

Разрешение некоторых конфликтов при помощи множеств $\mathrm{FOLLOW}(A)$

- Свёртка по правилу $A \to \alpha$ является правильным действием LR-анализатора, только если очередной входной символ принадлежит $\mathrm{FOLLOW}(A)$.
- Пусть пункт $[A \to \alpha\cdot]$ $(A \neq S')$ принадлежит состоянию $\mathtt{I}$ LR(0)-автомата.
- Тогда достаточно положить $\mathrm{ACTION}(\mathtt{I}, \mathtt{a}) := (\otimes\text{номер}(A \to \alpha))$ лишь для всех $\mathtt{a} \in \mathrm{FOLLOW}(A)$.

Алгоритм построения SLR(1)-анализатора

Вход. Расширенная грамматика $G = (\Sigma, \Gamma, P, S')$.
Выход. Таблица LR-анализа для грамматики G.

1. построить LR(0)-автомат $\mathcal{A}_G = (Q, \Sigma \cup \Gamma, \delta, \mathtt{I}_0, Q)$;
2. построить $\textsc{follow}(A)$ для каждого $A \in \Gamma$;
3. **для каждого** $\mathtt{I} \in Q$
4. **для каждого** $\mathtt{i} \in \mathtt{I}$
5. **если** $(\mathtt{i} = [S' \to S\cdot])$ $\textsc{action}(\mathtt{I}, \dashv) := \checkmark$;
6. **иначе если** $(\mathtt{i} = [A \to \alpha\cdot])$
7. **для каждого** $\mathtt{a} \in \textsc{follow}(A)$
8. $\textsc{action}(\mathtt{I}, \mathtt{a}) := (\otimes\text{номер}(A \to \alpha))$;
9. **иначе если** $(\mathtt{i} = [A \to \beta_1 \cdot \mathtt{a}\beta_2])$
10. $\textsc{action}(\mathtt{I}, \mathtt{a}) := (\leftarrow \delta(\mathtt{I}, \mathtt{a}))$;
11. **иначе** (т. е. $\mathtt{i} = [A \to \beta_1 \cdot B\beta_2]$)
12. $\textsc{goto}(\mathtt{I}, B) := \delta(\mathtt{I}, B)$

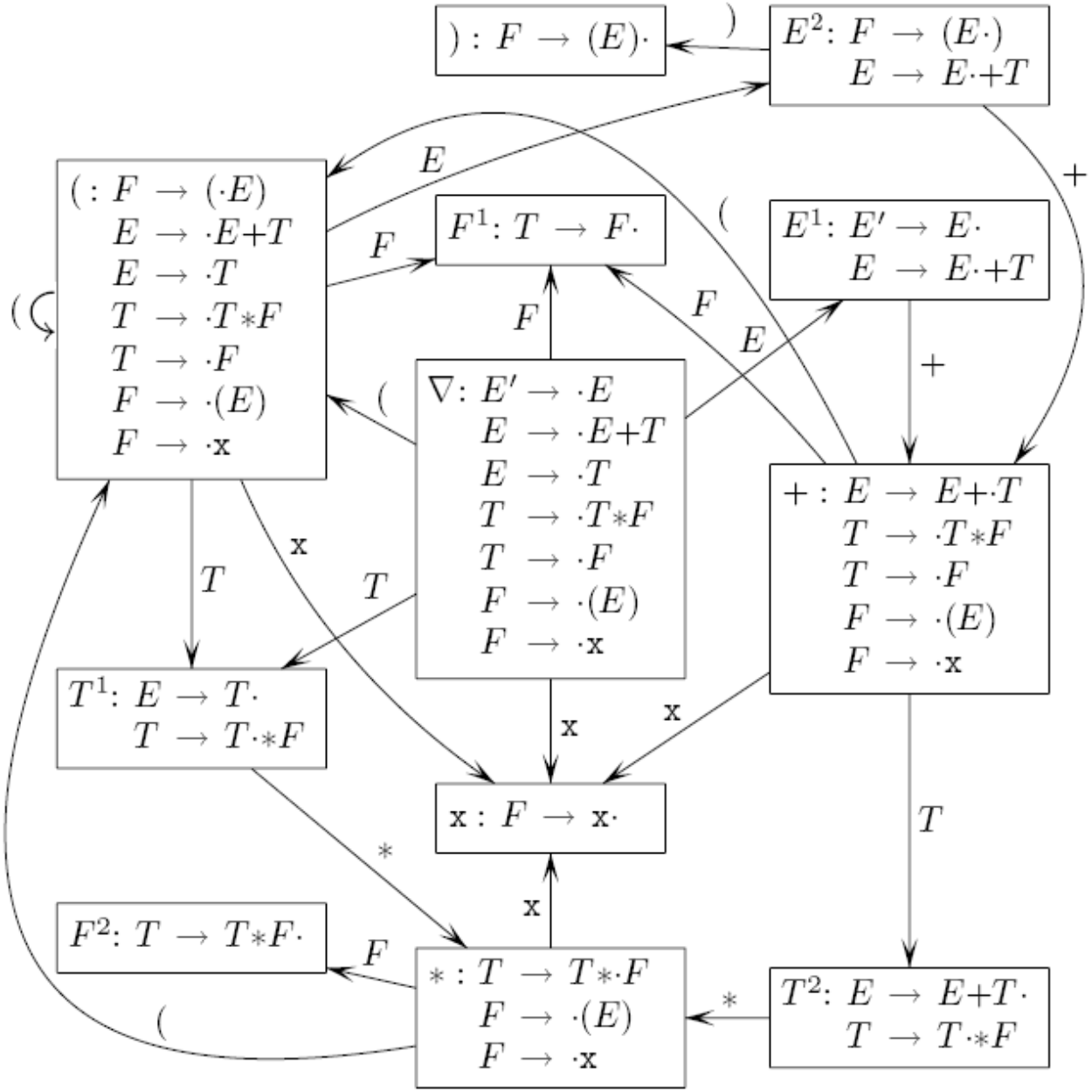
SLR(1)-грамматика: определение и пример

Грамматика называется *SLR(1)-грамматикой*, если в таблице ACTION, построенной алгоритмом на предыдущем слайде, нет конфликтов (т. е. в каждой клетке не более одной записи).

Пример. G_2 : (0) $S' \to S$, (1) $S \to \mathtt{a}D\mathtt{b}$, (2) $D \to D\mathtt{b}$, (3) $D \to \mathtt{b}$.
$\mathrm{FOLLOW}(S) = \{\dashv\}$, $\mathrm{FOLLOW}(D) = \{\mathtt{b}\}$.

	ACTION			GOTO	
	a	b	$\dashv$	S	D
S			✓		
D		$\leftarrow \mathtt{b}^2$			
a		$\leftarrow \mathtt{b}^1$			D
$\mathtt{b}^1$		$\otimes 3$			
$\mathtt{b}^2$		$\otimes 2$	$\otimes 1$		
∇	$\leftarrow \mathtt{a}$			S	

(0) $E' \to E$, (1) $E \to E + T$, (2) $E \to T$, (3) $T \to T * F$,
(4) $T \to F$, (5) $F \to (E)$, (6) $F \to \mathtt{x}$.



Пример построения SLR(1)-анализатора: окончание

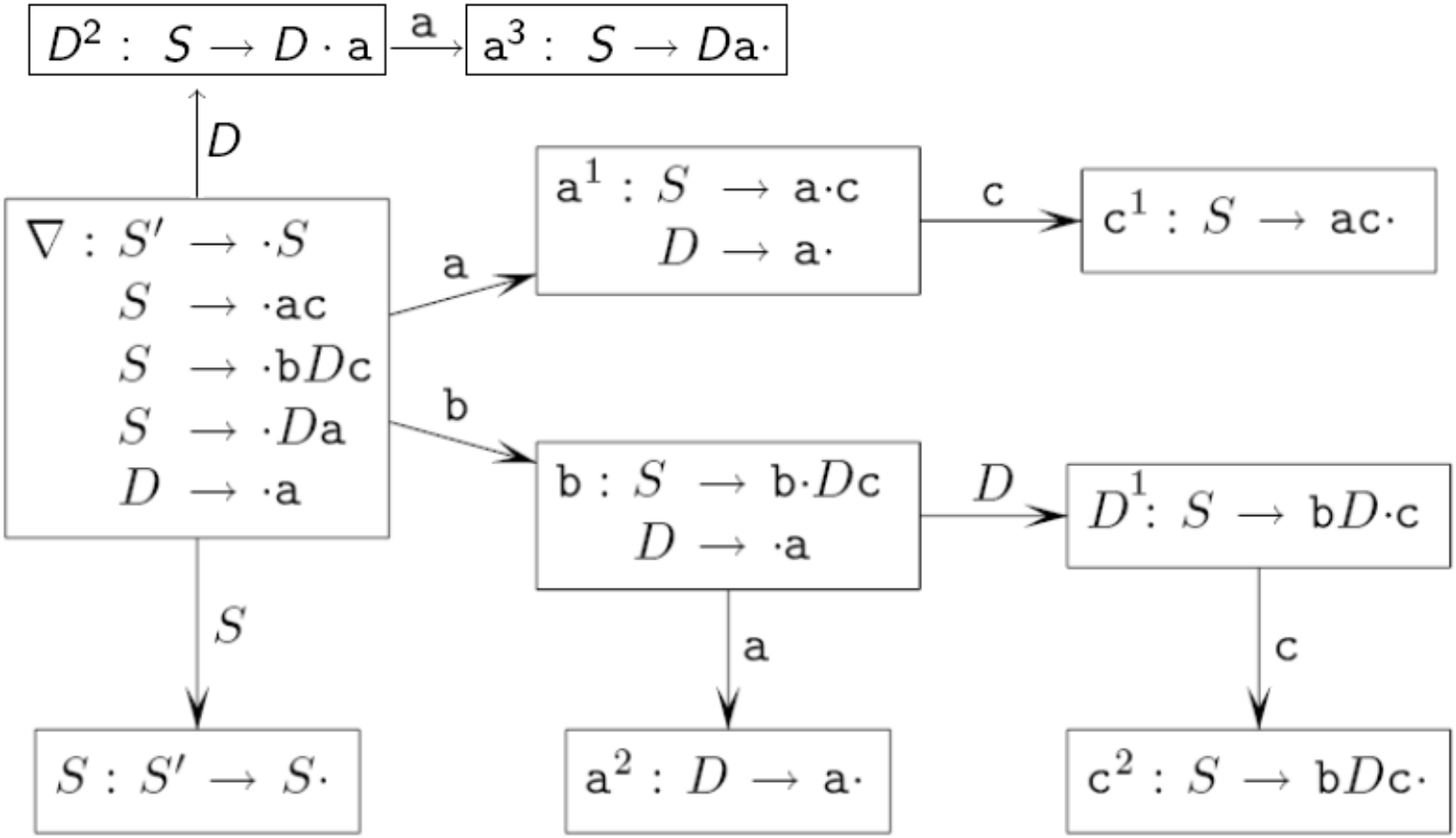
$\textsc{follow}(E) = \{+,), \dashv\}$, $\textsc{follow}(T) = \textsc{follow}(F) = \{*, +,), \dashv\}$.

	ACTION						GOTO		
	$+$	$*$	x	$($	$)$	$\dashv$	E	T	F
E^1	$\leftarrow +$					$\checkmark$			
T^1	$\otimes 2$	$\leftarrow *$			$\otimes 2$	$\otimes 2$			
F^1	$\otimes 4$	$\otimes 4$			$\otimes 4$	$\otimes 4$			
$($			$\leftarrow$ x	$\leftarrow ($			E^2	T^1	F^1
x	$\otimes 6$	$\otimes 6$			$\otimes 6$	$\otimes 6$			
$+$			$\leftarrow$ x	$\leftarrow ($				T^2	F^1
$*$			$\leftarrow$ x	$\leftarrow ($					F^2
E^2	$\leftarrow +$				$\leftarrow)$				
T^2	$\otimes 1$	$\leftarrow *$			$\otimes 1$	$\otimes 1$			
F^2	$\otimes 3$	$\otimes 3$			$\otimes 3$	$\otimes 3$			
$)$	$\otimes 5$	$\otimes 5$			$\otimes 5$	$\otimes 5$			
∇			$\leftarrow$ x	$\leftarrow ($			E^1	T^1	F^1

План

Пример не SLR(1)-грамматики

$G_4 : S' \to S,\ S \to \mathtt{ac} | \mathtt{b}D\mathtt{c} | D\mathtt{a},\ D \to \mathtt{a}.$



SLR(1)-анализатор для G_4 содержит конфликт в состоянии $\mathtt{a}^1$, поскольку $\mathrm{FOLLOW}(D) = \{\mathtt{c}, \mathtt{a}\}$. Разрешение конфликта: свёртку делать, если очередной входной символ a, но не c.

LR(1)-пункты

Пусть $G = (\Sigma, \Gamma, P, S')$ — расширенная грамматика.

- LR(1)-пункт: $[A \to \beta_1 \cdot \beta_2, \mathtt{a}]$, где
  - $[A \to \beta_1 \cdot \beta_2]$ — LR(0)-пункт, называемый *ядром* этого LR(1)-пункта,
  - $\mathtt{a}$ — терминал или символ $\dashv$.
- LR(1)-пункт $[A \to \beta_1 \cdot \beta_2, \mathtt{a}]$ допустим для активного префикса $\alpha\beta_1$, если существует (правый) вывод

$$S' \Rightarrow^* \alpha A w \Rightarrow \alpha\beta_1\beta_2 w \Rightarrow^* uw$$

  и цепочка $w \dashv$ начинается символом $\mathtt{a}$.
- *Автоматом LR(1)-пунктов* грамматики G назовём ε-НКА $\mathcal{I}^{(1)}_G = (Q,\ \Sigma \cup \Gamma,\ \delta,\ \{\mathtt{i}_0\},\ Q)$, где
  - Q — множество всех LR(1)-пунктов грамматики G,
  - $\mathtt{i}_0 = [S' \to \cdot S, \dashv]$,
  - отношение переходов состоит из всех базисных переходов вида $([A \to \beta_1 \cdot X\beta_2, \mathtt{a}],\ X,\ [A \to \beta_1 X \cdot \beta_2, \mathtt{a}])$ и всех ε-переходов вида $([A \to \beta_1 \cdot B\beta_2, \mathtt{a}],\ \varepsilon,\ [B \to \cdot\beta, \mathtt{b}])$, где $\mathtt{b} \in \mathrm{FIRST}(\beta_2\mathtt{a})$ и при построении $\mathrm{FIRST}(\beta_2\mathtt{a})$ мы считаем $\dashv$ терминалом.
- Для $\mathcal{I}^{(1)}_G$ верны основная теорема LR-анализа и её следствия.

LR(1)-автомат

- ДКА (неполный) $\mathcal{A}_G^{(1)}$, построенный по ε-НКА $\mathcal{I}_G^{(1)}$ алгоритмом из лекции 2, будем называть *LR(1)-автоматом* грамматики G.

  Пусть M — произвольное множество LR(1)-пунктов расширенной грамматики G.
- Положим, что $\textsc{closure}_1(M)$ есть минимальное по включению множество LR(1)-пунктов такое, что
  - $M \subseteq \textsc{closure}_1(M)$ и
  - $[A \to \beta_1 \cdot B\beta_2, \mathsf{a}] \in \textsc{closure}_1(M)$ влечёт $[B \to \cdot\gamma, \mathsf{b}] \in \textsc{closure}_1(M)$ для каждого правила вывода вида $B \to \gamma$ и каждого $\mathsf{b} \in \textsc{first}(\beta_2\mathsf{a})$.
- Пусть X — терминал или нетерминал грамматики G. Положим

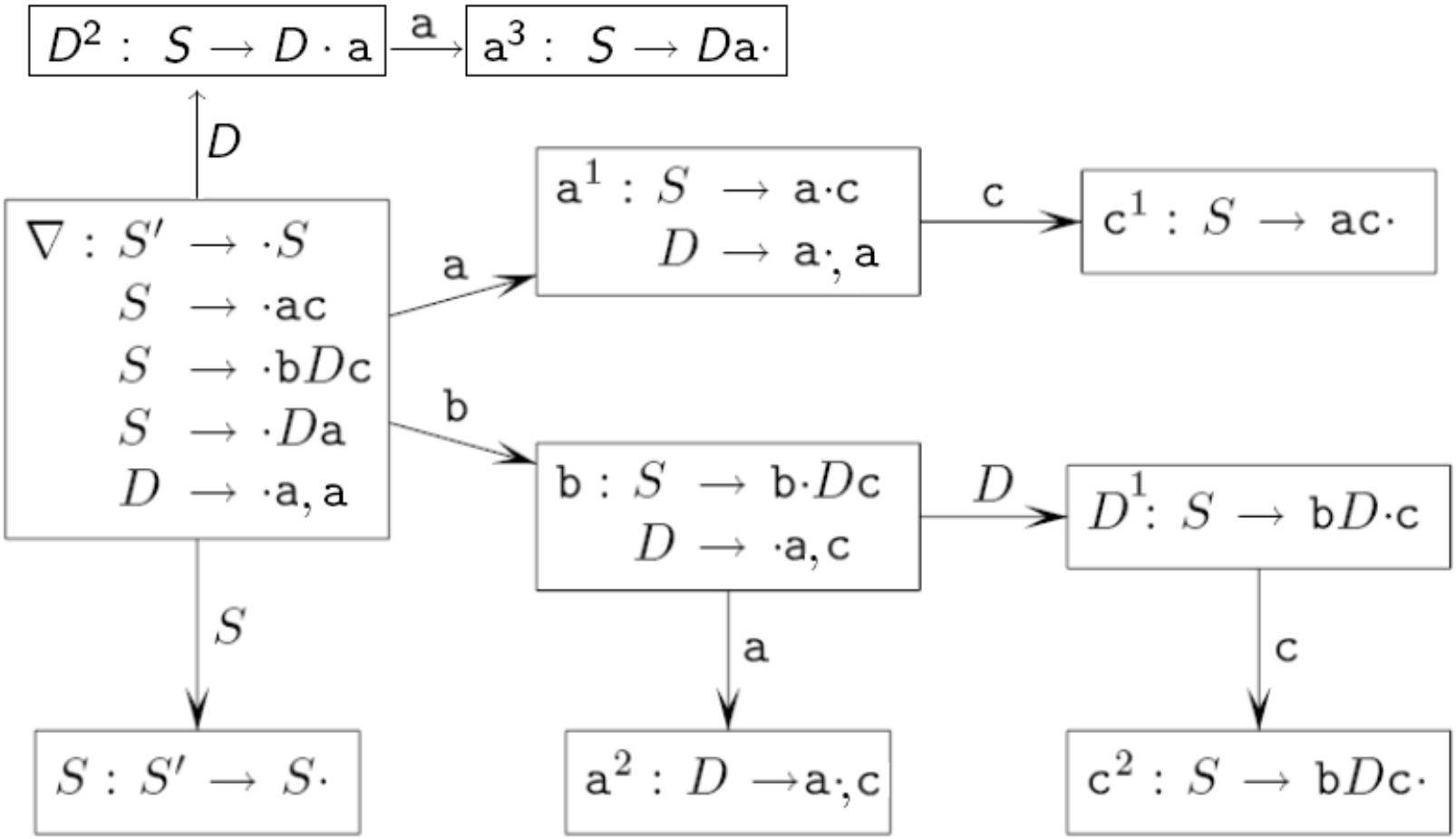
  $$\textsc{goto}_1(M, X) = \textsc{closure}_1(\{[A \to \beta_1 X \cdot \beta_2, \mathsf{a}] \mid [A \to \beta_1 \cdot X\beta_2, \mathsf{a}] \in M\}).$$
- Алгоритм построения LR(1)-автомата.

Пример построения LR(1)-автомата

G_4 : (0) $S' \to S$, (1) $S \to \mathtt{ac}$, (2) $S \to \mathtt{b}D\mathtt{c}$, (3) $S \to D\mathtt{a}$, (4) $D \to \mathtt{a}$.

Переходы в $\mathcal{I}^{(1)}_{G_4}$ имеют вид $([A \to \beta_1 \cdot X\beta_2, \mathtt{a}],\ X,\ [A \to \beta_1 X \cdot \beta_2, \mathtt{a}])$ и $([A \to \beta_1 \cdot B\beta_2, \mathtt{a}],\ \varepsilon,\ [B \to \cdot\beta, \mathtt{b}])$, где $\mathtt{b} \in \mathrm{FIRST}(\beta_2\mathtt{a})$.

На этом рисунке опущенный второй компонент в LR(1)-пунктах есть $\dashv$.



Алгоритм построения LR(1)-анализатора. Определение LR(1)-грамматики

Вход. Расширенная грамматика $G = (\Sigma, \Gamma, P, S')$.
Выход. Таблица LR-анализа для грамматики G.

1. построить LR(1)-автомат $\mathcal{A}^{(1)}_G = (Q, \Sigma \cup \Gamma, \delta, \mathtt{I}_0, Q)$;
2. **для каждого** $\mathtt{I} \in Q$
3. **для каждого** $\mathtt{i} \in \mathtt{I}$
4. **если** $(\mathtt{i} = [S' \to S\cdot, \dashv])$ $\textsc{action}(\mathtt{I}, \dashv) := \checkmark$;
5. **иначе если** $(\mathtt{i} = [A \to \alpha\cdot, \mathtt{a}])$
6. $\textsc{action}(\mathtt{I}, \mathtt{a}) := (\otimes\text{номер}(A \to \alpha))$;
7. **иначе если** $(\mathtt{i} = [A \to \beta_1 \cdot \mathtt{a}\beta_2, \mathtt{c}])$
8. $\textsc{action}(\mathtt{I}, \mathtt{a}) := (\leftarrow \delta(\mathtt{I}, \mathtt{a}))$;
9. **иначе** (т. е. $\mathtt{i} = [A \to \beta_1 \cdot B\beta_2, \mathtt{c}]$)
10. $\textsc{goto}(\mathtt{I}, B) := \delta(\mathtt{I}, B)$

Грамматика называется *LR(1)-грамматикой*, если в построенной этим алгоритмом таблице $\textsc{action}$ нет конфликтов (т. е. в каждой клетке не более одной записи).

Пример построения LR(1)-анализатора

$G_4 : (0)\ S' \to S,\ (1)\ S \to \mathtt{ac},\ (2)\ S \to \mathtt{b}D\mathtt{c},\ (3)\ S \to D\mathtt{a},\ (4)\ D \to \mathtt{a}.$

	ACTION				GOTO	
	a	b	c	$\dashv$	S	D
S				✓		
D^1			$\leftarrow \mathtt{c}^2$			
D^2	$\leftarrow \mathtt{a}^3$					
$\mathtt{a}^1$	$\otimes 4$		$\leftarrow \mathtt{c}^1$			
$\mathtt{a}^2$			$\otimes 4$			
$\mathtt{a}^3$				$\otimes 3$		
b	$\leftarrow \mathtt{a}^2$					D^1
$\mathtt{c}^1$				$\otimes 1$		
$\mathtt{c}^2$				$\otimes 2$		
∇	$\leftarrow \mathtt{a}^1$	$\leftarrow \mathtt{b}$			S	D^2

План

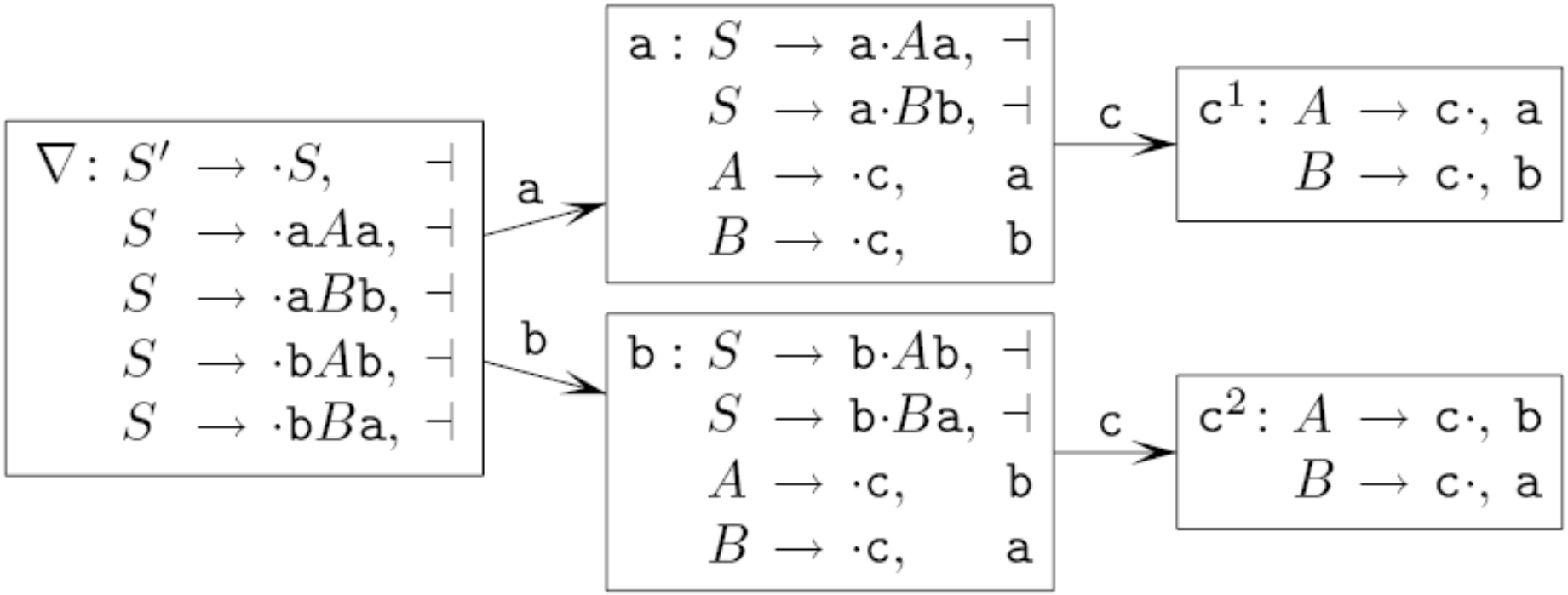
LARL(1)-анализ

- Для языка программирования Паскаль LR(1)-автомат имеет несколько тысяч состояний, тогда как LR(0)-автомат — несколько сотен состояний.
- LALR — Look Ahead LR, LR-анализ с предпросмотром.
- *LALR(1)-автомат* грамматики получается из LR(1)-автомата этой грамматики объединением всех состояний, имеющих одно и то же множество ядер LR(1)-пунктов, в одно состояние.
- Для LR(1)-грамматики LALR(1)-автомат не имеет конфликтов перенос-свёртка.
  - Пусть $[A \to \alpha\cdot, \mathrm{a}]$ и $[B \to \beta_1 \cdot \mathrm{a}\beta_2, \mathrm{b}]$ принадлежат одному состоянию LALR(1)-автомата.
  - Тогда $[A \to \alpha\cdot, \mathrm{a}]$ и $[B \to \beta_1 \cdot \mathrm{a}\beta_2, \mathrm{c}]$ (для некоторого c) принадлежат одному состоянию LR(1)-автомата.

  (Но конфликты свёртка-свёртка могут появиться.)
- Грамматика называется *LALR(1)-грамматикой*, если нет конфликтов в таблице ACTION, построенной алгоритмом на слайде 25, но по LALR(1)-автомату этой грамматики.

Пример LR(1)-, но не LALR(1)-грамматики

$G_5 : S' \to S,\ S \to \mathtt{a}A\mathtt{a}|\mathtt{a}B\mathtt{b}|\mathtt{b}A\mathtt{b}|\mathtt{b}B\mathtt{a},\ A \to \mathtt{c},\ B \to \mathtt{c}.$



$\mathtt{c}^{1,2} = \mathtt{c}^1 \cup \mathtt{c}^2 = \{[A \to \mathtt{c}\cdot, \mathtt{a}/\mathtt{b}], [B \to \mathtt{c}\cdot, \mathtt{a}/\mathtt{b}]\}.$

($[A \to \alpha \cdot \beta, \mathtt{a}/\mathtt{b}]$ — сокращение $[A \to \alpha \cdot \beta, \mathtt{a}],\ [A \to \alpha \cdot \beta, \mathtt{b}]$.)

Литература

Основная литература

- Замятин А. П., Шур А. М. Языки, грамматики, распознаватели: Учебное пособие. Екатеринбург : Изд-во Урал. ун-та, 2007 (электронный вариант книги — на `http://elar.usu.ru`, поиск).

Дополнительная литература

- Ахо А., Лам М., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструментарий. М.: ООО "И.Д. Вильямс", 2008.
- Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978.
- Мартыненко Б. К. Языки и трансляции: Учеб. пособие. СПб.: Издательство С.-Петербургского университета, 2004 (электронный вариант книги — на `http://www.math.spbu.ru/user/mbk`).