

3.3 Синтактичен анализ на безконтекстни езици

Синтактичният анализ е важна част от работата на всички компилатори на езиците за програмиране, тъй като в почти всички етапи на компилацията се изисква установяване на съответствие между зададените в програмата структури и тези, които са заложени в съответния алгоритмичен език. По същество това се свежда до отговор на въпроса, дали зададена входна дума може да се породи от граматиката на съответния език за програмиране. Най-често граматиката на езиците за програмиране се състои от обединението на автоматни и безконтекстни граматики.

Тъй като всяка безконтекстна граматика е еквивалентна на някой недетерминиран магазинен автомат, то задачата за синтактичен анализ се състои в построяването на съответния НМА и по работата му върху дадена входна дума, да се даде отговор дали тази дума се поражда от съответната граматика или не. По същество работата на НМА се състои в построяване на съответното дърво на извод на дадената дума.

Съществуват два основни алгоритъма за изграждане на дървета на извод.

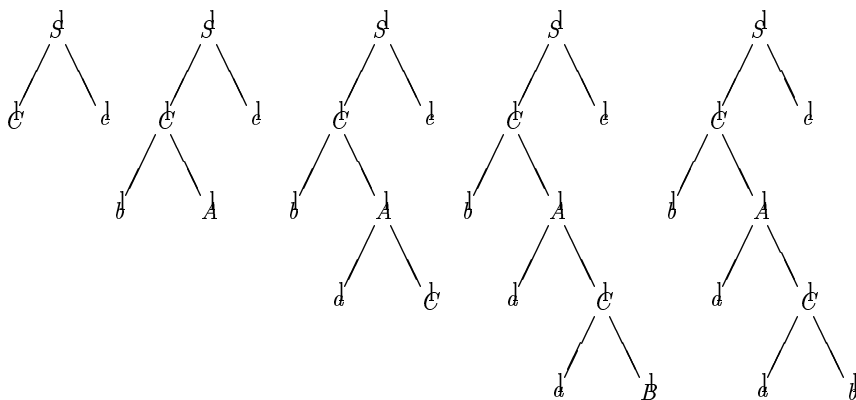
При синтактичен анализ "отгоре надолу", автоматът построява последователните конфигурации с дадената входна дума ω , като започва от началната конфигурация, в която $z_0 = S$ е корен на съответното дърво. Когато автоматът стигне до състояние, което не е крайно, но не може да продължи е необходимо да се върне на най-близкото предишно състояние, в което е могло да се направи друг избор. По този начин се обхождат всички възможни редици от конфигурации и се решава задачата за анализ на безконтекстните езици. В действителност, за по-голяма ефективност, на самите безконтекстни граматики се налагат допълнителни ограничения, с които чувствително се ускорява алгоритъмът за синтактичен анализ. Ясно е, че ако дадената граматика е еквивалентна на някои детерминиран магазинен автомат, то съответния алгоритъм е много по-ефективен.

При синтактичния анализ "отдолу нагоре" се възстановяват върховете на дървото на извод като се започне от листата a_1, a_2, \dots, a_s , където $\omega = a_1 a_2 \dots a_s$. Ако по този начин достигнем до корена на дървото, то ω може да се породни в съответната граматика, а в противен случай ω не е дума от зададения с граматиката език.

Ще илюстрираме двата алгоритъма, чрез следния пример.

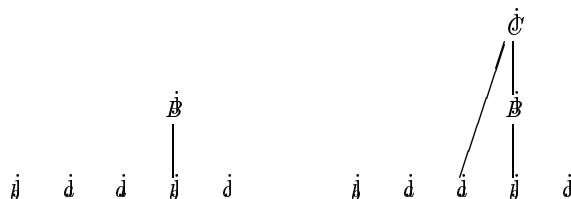
Пример 3.3.1 Нека Γ е безконтекстна граматика с правила $S \rightarrow Cc$, $C \rightarrow bA|aB$, $A \rightarrow a|aC$, $B \rightarrow b|bC$ и да построим дървото на извод по двата начина за думата $baabc$.

а) "отгоре - надолу" (Фиг. 3.3.1)

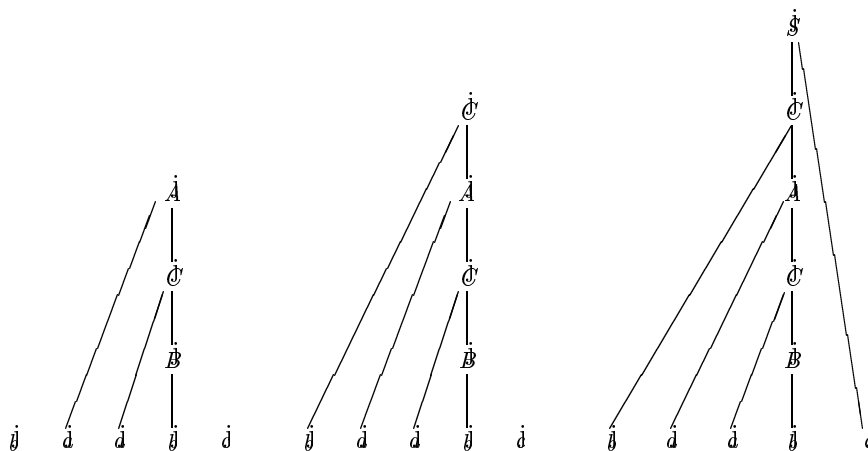


Фиг. 3.3.1

б) "отдолу - нагоре" (Фиг. 3.3.2 и 3.3.3)



Фиг. 3.3.2



Фиг. 3.3.3

З а д а ч и

1. Да се построят дърветата на извод на следните аритметични изрази:

а) $(a + b) * c$;

б) $c * (a + b)$;

в) $(a + b) * (b + c)$.

2. Да се построят дърветата на извод на следните думи :

а) $a^3b^2a^2b^2$;

б) $ab^2a^3b^4$;

в) a^4b^4

в граматиката със следните правила:

$$S \rightarrow a|aA, \quad A \rightarrow aA|bB|bb, \quad B \rightarrow bA.$$