Dual pushdown automata and context sensitive grammars

By GY. RÉVÉSZ

1. Introduction

In recent years, a number of generalizations of pushdown automata have been studied. The basic model of pushdown automata bears an equivalence relationship to context-free grammars as shown by Chomsky [1] and Schützenberger [2]. Gray, Harrison and Ibarra extended this model as they studied two-way pushdown automata [3] while Ginsburg, Greibach and Harrison introduced stack automata [4 and 5]. A stack automaton is essentially a pushdown automaton which is allowed to scan the inside of its pushdown store without having to erase, i.e., in a read only mode. Stack automata are closely related to context sensitive grammars [6], but they are not equivalent to them. (See e.g. in [7].)

In the present paper we offer a new model called dual pushdown automaton (DUPA), since it has two pushdown stores which are complementary to each other. This model can be motivated by a normal form of context sensitive grammars which we shall see later. It can be seen that dual pushdown automata are equivalent to context sensitive grammars and, which is the same, to linear bounded automata [8 and 9].

To every context sensitive grammar in normal form we can construct a DUPA that always performs the leftmost replacement(s) while parsing sentences of the given context sensitive language. This feature may be useful for parsing from left to right, which is of great importance in connection with the direct interpretation of algorithmic languages by machine (without translation) as suggested by Kalmár [10]. Namely, according to the concept of Kalmár's formula directed computer the execution of an algorithm written in a mathematical formula language proceeds as follows. The description of the algorithm, i.e., the program of the calculation is analysed from left to right and, whenever a syntactic unit is recognized, it is semantically interpreted. Naturally, for this purpose we need a suitable language where no back tracking is necessary for the syntactic analysis. It seems useful to treat this problem with the aid of context sensitive grammars even if we are concerned with context-free languages only.

In the present paper we discuss only the basic relation of dual pushdown automata to context sensitive grammars. The problem of left-to-right parsing with respect to a specific subclass of context sensitive (namely, unilateral context sensitive) grammars has been studied in [11] whose results can very likely be generalized for context sensitive grammars in normal form. However, the problem of transforming unfeasible grammars into suitable forms has not been solved yet in general. This problem is also related to the problem of simplifying given arbitrary dual pushdown automata.

2. Preliminaries

The set of words (including the empty word ε) over a finite set of symbols V will be denoted by V^*. Individual symbols will be denoted by small latin letters while words and sets of symbols by capitals.

Definition 1. A context sensitive grammar is a quadruple $G=(T, V, s, P)$, where T and V are finite sets of symbols, $T\subset V$, $s\in V-T$ and P is a finite set of ordered pairs — called rules — of the form $XqY\to XQY$, where $q\in V-T$ while X, Y and Q are in V^* and $Q\neq\varepsilon$ (i.e., Q non-empty).

Definition 2. A context sensitive grammar G is said to be in normal form if every rule in P is of the form $a\to b$ or $a\to bc$ or $ac\to bc$ or $ab\to ac$, where a, b and c are in V.

Definition 3. For a given context sensitive grammar G and two words A and $B\in V^*$, B is an immediate consequence of A (in symbols $A\Rightarrow B$), if there exists a rule $XqY\to XQY$ in P such that $A=UXqYZ$ and $B=UXQYZ$ for some $U, Z\in V^*$.

Definition 4. For a given context sensitive grammar and two words A and $B\in V^*$, B is derivable from A (in symbols $A\overset{*}{\Rightarrow}B$), if there exists a finite sequence of words $X_0, X_1, \dots, X_n$ each in V^* such that $A=X_0$, $B=X_n$ and $X_i\Rightarrow X_{i+1}$ for $0\leq i<n$. The sequence $X_0, X_1, \dots, X_n$ is then called a derivation of B from A with respect to G.

Definition 5. For a given context sensitive grammar G the set of words

$$L_G = \{W|s \overset{*}{\Rightarrow} W\}\cap T^*$$

is the language generated by G.

Two grammars are called weak-equivalent if they generate the same language.

A DUPA may be informally illustrated as in Fig. 1. Each move of the device is determined by the actual state of the finite state control and the topmost symbols in the two pushdown stores.

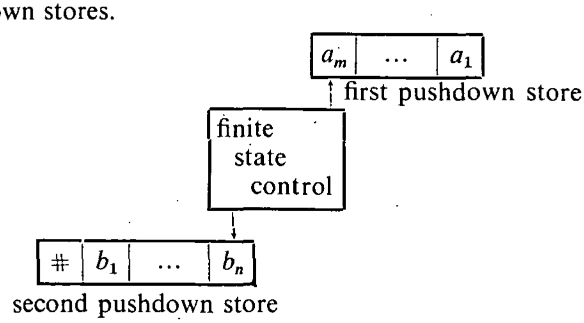

Fig. 1. Dual pushdown automaton

Each move consists of moving the two read-write heads at most one square right or left and writing a new symbol on one of the two read-write positions. The movements of the two read-write heads are coordinated such that only four types exist: $\{R, S, L, D\}$, i.e., right, stay, left, delete.

The DUPA is stopped, if it attempts to read a symbol from an empty pushdown store, i.e., if its first read-write head is to be positioned below the bottom of the corresponding pushdown store.

We now give the formal definition of a DUPA.

Definition 6. A dual pushdown automaton is an 8-tuple $A=(K, \Sigma, T, s, \#, \delta, p_1, F)$ where

(i) K is a finite nonempty set (of states),

(ii) Σ and T are finite nonempty sets (of symbols), $T\subset\Sigma$, $s\in\Sigma\dot{-}T$,

(iii) $\#$ is the left endmarker: $\#\notin\Sigma$,

(iv) δ is a mapping from $K\times(\Sigma\cup\{\#\})\times\Sigma$ into the finite subsets of $K\times\Sigma\times\times\{R, S, L, D\}$ such that $(p', x', L)\notin\delta(p, \#, x)$ and $(p', x', D)\notin\delta(p, \#, x)$ for any p, p', x, x'.

(v) $p_1\in K$ (initial state) and $F\subseteqq K$ (final states).

If the mapping δ is unique then A is deterministic otherwise it is nondeterministic.

Definition 7. A configuration of a DUPA is any element of the set $K\times\#\Sigma^*!\Sigma^*$, where $!\notin\Sigma$.

Definition 8. Let $\vdash$ be the binary relation defined on the set of configurations as follows.

For arbitrary $a\in(\Sigma\cup\{\#\})$, and $b, c\in\Sigma$ and $X\in(\Sigma\cup\{\#\})^*$, $Y\in\Sigma^*$

$$\begin{aligned}
(p, Xa!bY)&\vdash(p', Xac!Y) && \text{if} \quad (p', c, R)\in\delta(p, a, b),\\
(p, Xa!bY)&\vdash(p', Xa!cY) && \text{if} \quad (p', c, S)\in\delta(p, a, b),\\
(p, Xa!bY)&\vdash(p', X!cbY) && \text{if} \quad (p', c, L)\in\delta(p, a, b),\\
(p, Xa!bY)&\vdash(p', X!cY) && \text{if} \quad (p', c, D)\in\delta(p, a, b).
\end{aligned}$$

Definition 9. Let $\models$ be the transitive closure of $\vdash$, i.e., for configurations z and z', $z\models z'$ if there exists a sequence of configurations $z_0, z_1, \dots, z_n$ such that $z_0=z$, $z_n=z'$ and $z_i\vdash z_{i+1}$ for $0\leqq i<n$.

Definition 10. A word $W\in\Sigma^*$ is accepted by a DUPA if $(p_1, \#!W)\models(p_f, \#s!)$ for some $p_f\in F$.

Definition 11. The set of all terminal words ($W\in T^*$) accepted by a DUPA is called the language accepted by it.

3. The relationship of DUPA to context sensitive grammars

Theorem 1. The language accepted by a DUPA can be generated by a context sensitive grammar in normal form.

Proof. To each DUPA we construct a context sensitive grammar as follows. Let $a_i\in V$ for every $a_i\in\Sigma$. Moreover to every pair (p_j, a_i) in $K\times\Sigma$ a new element

$a_i^{(j)} \in V$ will be defined. The set of rules will be defined such that

$$a_k^{(i)} \to a_r^{(j)} \in P \quad \text{if} \quad (p_i, a_k, R) \in \delta(p_j, \#, a_r)$$
$$\text{or} \quad (p_i, a_k, S) \in \delta(p_j, \#, a_r),$$
$$a_l a_k^{(i)} \to a_l a_r^{(j)} \in P \quad \text{if} \quad (p_i, a_k, R) \in \delta(p_j, a_l, a_r)$$
$$\text{or} \quad (p_i, a_k, S) \in \delta(p_j, a_l, a_r),$$
$$a_k^{(i)} a_r \to a_l^{(j)} a_r \in P \quad \text{if} \quad (p_i, a_k, L) \in \delta(p_j, a_l, a_r),$$
$$a_k^{(i)} \to a_l^{(j)} a_r \in P \quad \text{if} \quad (p_i, a_k, D) \in \delta(p_j, a_l, a_r).$$

In addition to that

$$a_k^{(1)} \to a_k \in P \text{ for every } a_k \in T,$$

$$\left.\begin{array}{l} a_l^{(i)} a_r \to a_l a_r^{(i)} \in P \\ a_l a_r^{(i)} \to a_l^{(i)} a_r \in P \end{array}\right\} \text{ for every } i, l, r$$

and

$$s \to a_r^{(j)} \in P \quad \text{if} \quad (p_f, s, R) \in \delta(p_j, \#, a_r) \text{ for}$$

some $p_f \in F$.

It can be easily verified that each word accepted by the DUPA can be generated by the grammar, if we follow the way of accepting the given word in reversed order.

On the other hand, to each word generated by the grammar a sequence of moves of the DUPA can be specified that corresponds to the reversed derivation of the given word.

Some of the rules of the grammar constructed above are of the form $ab \to cd$, which is not allowed in the normal form (see Definition 2.), but each of these can be replaced by three rules of the form $ab \to ab'$, $ab' \to cb'$ and $cb' \to cd$.

Theorem 2. The language generated by a context sensitive grammar is accepted by a DUPA having one internal state only.

Proof. It is known that each context sensitive grammar is weak-equivalent to one in normal form [9]. Thus, we have to consider context sensitive grammars in normal form only. The corresponding DUPA will be defined as follows:

Let $\Sigma = V$ and the mapping δ defined such that if $a_k \to a_r \in P$ then $(p_1, a_k, S) \in \delta(p_1, a_l, a_r)$ for every $a_l \in V$, if $a_k \to a_l a_r \in P$ then $(p_1, a_k, D) \in \delta(p_1, a_l, a_r)$, if $a_l a_k \to a_l a_r \in P$ then $(p_1, a_k, S) \in \delta(p_1, a_l, a_r)$, if $a_k a_r \to a_l a_r \in P$ then $(p_1, a_k, L) \in \delta(p_1, a_l, a_r)$. Moreover

$$(p_1, a_r, R) \in \delta(p_1, a_l, a_r)$$
$$(p_1, a_r, R) \in \delta(p_1, \#, a_r)$$
$$(p_1, a_l, L) \in \delta(p_1, a_l, a_r)$$

for every a_l, a_r in V.

It can be seen again that each word generated by the grammar is accepted by the DUPA and vica versa.

Corollary. Each DUPA is equivalent to a DUPA having one internal state only.

Thus, we can say that the finite state control of the DUPA is superfluous since it can be replaced by a single state control.

Naturally the number of internal states will be decreased at the cost of increasing the number of auxiliary symbols. The construction of a minimal (in some sense) DUPA to a given context sensitive grammar is an open question.

Deterministic DUPA can be easily implemented and used for practical purposes, but it is to be ensured that the language to be recognized is of suitable structure. Usually the grammar generating the language must be transformed into an appropriate form (if possible) and the transformed grammar is more complex than the original one. These questions are not discussed here, since they are not sufficiently elaborated yet.

References

[1] Chomsky, N., Context-free grammars and pushdown storage, *Quart. Progress Report*, v. 65, Research Laboratory of Electronics, MIT, 1962.

[2] Schützenberger, M. P., On context-freee languages and push-down automata, *Information and Control*, v. 6, 1963, pp. 246—264.

[3] Gray, J. N., M. A. Harrison, O. H. Ibarra, Two-way pushdown automata, *Information and Control*, v. 11, 1967, pp. 30—70.

[4] Ginsburg, S., S. A. Greibach, M. A. Harrison, One-way stack automata, *J. Assoc. Comput. Mach.*, v. 14, 1967, pp. 389—418.

[5] Ginsburg, S., S. A. Greibach, M. A. Harrison, Stack automata and compiling, *J. Assoc. Comput. Mach.*, v. 14, 1967, pp. 172—201.

[6] Hopcroft, J. E. & J. D. Ullman, Sets accepted by one-way stack automata are context sensitive, *Information and Control*, v. 13, 1968, pp. 114—133.

[7] Aho, A. V. & J. D. Ullman, The theory of languages, *Math. Systems Theory*, v. 2, 1968, pp. 97—125.

[8] Landweber, P. S., Three theorems on phrase structure grammars of type 1. *Information and Control*, v. 6, 1963, pp. 131—136.

[9] Kuroda, S. Y., Classes of languages and linear bounded automata, *Information and Control*, v. 7, 1964, pp. 207—223.

[10] Kalmár, L., On a digital computer which can be programmed in a mathematical formula language. *II. Hungarian Mathematical Congress*, v. 2, Section V. Budapest 1960, pp. 3—16.

[11] Révész, Gy., Syntactic analysis and unilateral context sensitive grammars, *Studia Sci. Math. Hungar.*, v. 4, 1969, pp. 267—278.

(Received Oct. 30, 1970)