THE EQUIVALENCE OF REDUCING TRANSITION LANGUAGES AND DETERMINISTIC LANGUAGES

Mario Schkolnick

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

July 1973

This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-73-C-0074) and is monitored by the Air Force Office of Scientific Research. This document has been approved for public release and sale; its distribution is unlimited.

ABSTRACT

The class of reducing transition languages introduced by Eickel, Paul, Bauer and Samelson was shown by Morris to be a proper superclass of the simple precedence languages. In this paper we extend this result showing that in fact, the first class is equivalent to the class of deterministic context free languages.

I. INTRODUCTION

The relationship between the class of reducing transition languages, introduced by Eickel, Paul, Bauer and Samelson [1], and the class of simple precedence languages, introduced by Wirth and Weber [2], was studied by Morris [3]. He showed that the latter class is a proper subset of the former and hinted that this first class was a proper subset of the deterministic context free languages. In this paper we show that the canonical grammar for a deterministic pushdown automata [4] is a reducing transition grammar, thus showing another characterization of deterministic languages. It is interesting to note that this is the earliest class of grammars described in the literature known to the author which characterizes the class of deterministic languages.

In Section II we present some basic definitions which we use in the paper. Section III is devoted to the proof that (1) the class of reducing transition grammars contains the class of weakly invertible, normal 2 form, simple mixed strategy precedence grammars, and (2) the canonical grammar for a DPDA is a member of this last class.

II. DEFINITIONS

We assume the reader is familiar with the basic concepts for context free languages. Any term not defined here may be found in [4,5]. A proper context free grammar $G = (V,V_T,S,P)$ is a reduced, Λ-free, cycle-free context free grammar. V is the vocabulary, $V_T \subseteq V$ is the set of terminals, $S \in V_N = V-V_T$ is the start symbol, $P \subseteq V_N \times V^+$ is the set of

productions, V_N is the set of nonterminals. $V^+ = V^* - \{\Lambda\}$. Elements of V_N will be denoted by capital letters near the front of the alphabet A,B,C,D,..., elements of V_T will be denoted by lower case letters near the front of the alphabet, elements of V will be denoted by capital letters near the end of the alphabet R,S,T,U,V,W,X,Y,Z and lower case letters near the end of the alphabet will denote words in V^*. The empty word is denoted by Λ.

A grammar $G = (V,V_T,S,P)$ is in 2 normal form if $A \to x$ in P implies $|x| \leq 2$. We will now recall the definition of precedence relations between symbols of V. First we describe complete left and right sets. This definition is the same as in [3]. Let $G = (V,V_T,S,P)$ be a proper context free grammar.

$\lambda'(U) = \{W \mid U \overset{+}{\Rightarrow} Wx \text{ for some } x \in V^*\}$
$\rho'(U) = \{W \mid U \overset{+}{\Rightarrow} xW \text{ for some } x \in V^*\}$
$\eta(U) = \{W \mid U \overset{+}{\Rightarrow} W\}$

Note that $U \in \lambda'(U)$ and $U \in \rho'(U)$. Also, since G is proper, $U \notin \eta(U)$ and the derivation $U \overset{+}{\Rightarrow} W$ involves only productions with righthand side of length one. We will call these productions chain rules, and derivations which involve chain rules only will be called chains.

Let $A = X_{j+1} \Rightarrow X_j \Rightarrow X_{j-1} \Rightarrow \ldots \Rightarrow X_1 \Rightarrow X_0 = Z$ and
$B = Y_{j+1} \Rightarrow Y_j \Rightarrow Y_{j-1} \Rightarrow \ldots \Rightarrow Y_1 \Rightarrow Y_0 = Z$

be two chains and $A \neq B$. Let k be the smallest integer such that $X_k = Y_k$ but $X_{k+1} \neq Y_{k+1}$. Then X_k is called the link of both chains. We also need the definitions of left and right sets:

$\lambda(U) = \{W \mid U \overset{+}{\Rightarrow} Wx\}$
$\rho(U) = \{W \mid U \overset{+}{\Rightarrow} xW\}$

Note that $\lambda(U) \cup \{U\} = \lambda'(U)$ and $\rho(U) \cup \{U\} = \rho'(U)$

Finally, let $\sigma'(U) = \lambda'(U) \cap V_T$.

The precedence relations are defined as:

$X \doteq Y$ iff $A \to xXYy$ is in P
$X \lessdot Y$ iff $A \to xXBy$ is in P and $Y \in \lambda(B)$

$X \gtrdot a$ $A \to xBYy$ is in P, $X \in P(B)$, $a \in \lambda'(Y)$

Note the relation $\gtrdot$ is defined on $V \times V_T$.

A grammar G is simple mixed strategy precedence (SMSP) if the following conditions hold:

M1: $(\lessdot \cup \doteq) \cap \gtrdot = \phi$

M2: If $A \to xXYy$ and $B \to Yy$ are productions, then $(X,B) \notin \lessdot \cup \doteq$

M3: If $A \to x$ and $B \to x$ are productions and $A \neq B$, then $I(A) \cap I(B) = \phi$, where $I(C) = \{X \mid (X,C) \in \lessdot \cup \doteq\}$

These conditions will be referred as M1, M2 and M3 respectively. We will say that a SMSP grammar is weakly invertible if the following condition is satisfied:

M4: If $A \to x$ and $B \to x$ are productions and $|x| > 1$, then $A = B$.

Thus, a weakly invertible grammar is uniquely invertible up to chain rules.

Let $\mathcal{T}$ denote the class of weakly invertible, 2 normal form, SMSP grammars. Any grammar in $\mathcal{T}$ satisfies M1-M4 and the additional condition $A \to x$ in P implies $1 \leq |x| \leq 2$. The class of languages generated by grammars from $\mathcal{T}$ will be denoted by $L(\mathcal{T})$.

Our last set of definitions has to do with the reducing transition grammars. We will follow [3]. Let $G = (V,V_T,S,P)$ be a proper grammar in normal 2 form. Let $V'_T = V_T \cup \{\#\}$ where # is a new symbol not in V. Let $V' = V \cup \{\#\}$. A reducing transition table will be a mapping $\Phi : V' \times V' \times V'_T \to (\{1,2\} \times V) \cup \{3\}$. The elements of the table will be written as $(A,B,C) \to (n,D)$ or $(A,B,C) \to (3)$ where $A,B \in V'$, $C \in V'_T$, $D \in V_N$ and $n \in \{1,2\}$. The term on the left of the arrow will be called a triple and the terms to the right of the arrow will be called doubles. We will now construct a reducing transition table M for the grammar G.

Let $R_i \to S_iT_i$ and $R_j \to S_jT_j$ be any two productions in P. Then:

if $R_j \in P'(S_i)$ then

(a) $\forall X \in \eta(T_i)$, $\forall b \in \sigma'(T_i)$, $(S_i,X,b) \to (1,T_i)$ is in M

(b) $\forall b \in \sigma'(T_i)$, $(S_i,T_i,b) \to (2,R_j)$ is in M

if $R_j \in \lambda'(T_i)$ then

(c) VX *i* 1l(Sj), Vb < a'(Tj), (S,,X,b) (I,Sj) is in M

(d) Vb < or'fTj), (Si,Sj,b) (3) is in M

if Rj (V(S) then

(e) VX < tl(Si), Vb < a'(Tj), (*,X,b) -»(I,Sj) is in M

(f) Vb < or'd,), (*,Sj,b) >(3) is in M

if Rj < P'(S) then

(g) VX / TUTj), (Sj,X,#) -(IJi) is in M

(h) (Si,Tj,tt) • (2,Rj) is in M

Finally,

(i) VX < Il(S), (tt,X,«) -••(1,S) is in M

If the mapping defined above is single valued, i.e., each triple has at most one associated double to it, then the grammar G is said to be a reducing transition grammar. The language generated by G is said to be a reducing transition language. Each entry in the table will be said to belong to class (a), (b),...,(h), (i) if it was generated by case (a), (b),...,(h), (i) respectively. For examples of these grammars the reader may consult [1] and [3].

III. RELATIONSHIP BETWEEN REDUCING TRANSITION LANGUAGES AND DETERMINISTIC LANGUAGES

Let D denote the class of deterministic context-free languages. Our first result is an immediate consequence of a result of Aho, Denning and Ullman [4,6].

Theorem *U* L(7) = D.

Proof: The reader is referred to the proof of Theorem 8.13 [4, vol. 2] or Theorem 3 [6]. The grammar obtained for a deterministic language is in the class 7. |

To get the desired characterization it is now sufficient to prove that the class L(7) is contained in the class of Reducing Transition Languages (RTL).

Theorem 2: L(7) <= RTL

Proof: Let G (7 be a proper grammar in normal 2 form. Assume that G is not a reducing transition grammar. Then, there are two entries in M with the same triples and with different doubles. As in [3] we call such entries culprit transitions. We first claim that we need to look only at a restricted number of cases.

Claim J_: Both culprit transitions are in one of the following classes: {(a), (b), (c)}, {(e)}, or {(g), <h»

Proof of Claim: Since ***n i*** V, both culprit transitions have to belong to either of {(a), (b), (c), (d)} or {(e), (f)} or {(g), (h)} or {(i)}. But, it follows from the way the transitions are defined, that (A,B,C) * (n,D), n < {1,2} implies B » C while (A,B,C) -> (3) implies B < C or B = C. Since G satisfies condition M1, we can. detach (d) from the first set and split the second set. Finally we note that any two transitions from either (d), (f), or (i) have unique righth'and sides so they cannot be culprit. Thus Claim 1 follows. |

To prove the theorem we have to analyze different cases depending on which class both culprit transitions came from. Let Rj SjTj, Rj SjTj generate one of the culprit transitions and R\ > S'jT'j, R'j • S'jT'j generate the other.

Case L (a) and (a). We have Sj = S'j, Tj / T'j, X < TItfj), X < IKT'j). Let C be the link of Tj U X and T'j U X. If C - T'j, 3D such that Tj U D, D •> T'j. But then (S'j,D) < i U = and R'j -> S'jT'j contradicting condition M2. A symmetric argument shows C * Tj. Thus, 3D,D' such that Tj ^> D, T'j U D',.D - C, D' « C. But then Sj < 1(D), S'j < I(D') which contradicts M3.

Case 2. (a) and (b). We have Sj = S'j, T'j « X < TI(Tj). Then Tj U T'j and we find a contradiction to M2 as in Case 1.

Case 3. (a) and (c). We have Sj = S'j, S'j * Tj, X <TI(Tj), X CtUS'j). Also, R'j C V(T,) so that S'j < S'j, b (a'(T'j) so that S'j i b and Rj (P'(Sj), b (a'(Tj) so that Tj > b. Let

C be the link of $T_j \stackrel{+}{\Rightarrow} X$ and $S'_j \stackrel{+}{\Rightarrow} X$. If $C = T_j$ $\exists D$ such that $S'_j \stackrel{*}{\Rightarrow} D$, $D \rightarrow T_j$. Then $S_j = S'_i \lessdot D$ which together with $R_j \rightarrow S_jT_j$ contradicts M2. If $C = S'_j$ then $T_j \stackrel{+}{\Rightarrow} S'_j$. But then $S'_j \gtrdot b$ which is a contradiction. Thus, $\exists D,D'$ such that $T_j \stackrel{*}{\Rightarrow} D$, $S'_j \stackrel{*}{\Rightarrow} D'$, $D \rightarrow C$, $D' \rightarrow C$. But then $(S_j,D) \in \lessdot \cup \doteq$, $(S'_i,D) \in \lessdot$ which contradicts condition M3.

Case 4. (b) and (b). We have $R_j \rightarrow S_jT_j$, $R'_j \rightarrow S_jT_j$, $R_j \neq R'_j$ which contradicts M4.

Case 5. (b) and (c). This is similar to Case 2. We have $T_j \in \eta(S'_j)$ so $S'_j \stackrel{+}{\Rightarrow} T_j$. But $S_j = S'_i \lessdot S'_j$ (because $R'_j \in \lambda'(T'_i)$. Thus we have $\exists D$ such that $S'_j \stackrel{*}{\Rightarrow} D$, $D \rightarrow T_j$ and $S_j \lessdot D$ which contradicts M2.

Case 6. (c) and (c). We have $S_i \lessdot S_j$, $S'_i \lessdot S'_j$, $S_i = S'_i$, $X \in \eta(S_j)$, $X \in \eta(S'_j)$, $S_j \neq S'_j$. Since $b \in \sigma'(T_j)$ and $b \in \sigma'(T'_j)$ we have that $S_j \lessdot b$ and $S'_j \lessdot b$. Thus, if the link of $S_j \stackrel{+}{\Rightarrow} X$ and $S'_j \stackrel{+}{\Rightarrow} X$ were S_j, we would have $S_j \gtrdot b$, a contradiction. Likewise S'_j cannot be the link. Thus $\exists D,D'$ such that $S_j \stackrel{*}{\Rightarrow} D$, $S'_j \stackrel{*}{\Rightarrow} D'$, $D \rightarrow C$, $D' \rightarrow C$ and $S_i = S'_i \in I(D)$, $S_i = S'_i \in I(D')$ contradicting M3.

Case 7. (e) and (e). This is exactly the same as Case 6 with # replacing S_i and S'_i.

The only remaining possibilities are (g) and (g), (g) and (h) and (h) and (h). But these are analyzed exactly as Cases 1, 2 and 4 with # instead of b. This concludes the proof of Theorem 2. ∎

Combining Theorems 1 and 2 we get

Theorem 3: The class of reducing transition languages is equivalent to the class of deterministic context free languages.

The main theorem in [3] now becomes a corollary of Theorem [3], since it is well known that simple precedence languages are a proper subset of the deterministic languages [7].

Corollary: The class of reducing transition languages properly includes the class of simple precedence languages.

Theorem 3 says that the class of reducing transition grammars is a very interesting one. It is very easy to parse sentences generated by it [1] and furthermore every deterministic language can be given one of these grammars.

It is interesting to relate the class of reducing transition grammars to other known classes of grammars. We note here the following result.

Theorem 4: The class of reducing transition grammars is incomparable with the class of normal 2 form (2-1) UI precedence grammars.

Proof: We exhibit two grammars each of which is in one class but not in the other.

G_1:	G_2:
$S \to Ad \mid Ce$	$S \to Ab \mid cC$
$A \to aB$	$A \to dD$
$B \to bE$	$D \to E$
$E \to c$	$E \to a$
$C \to aD$	$C \to Bb$
$D \to bc$	$B \to dE$

It is easy to verify that G_1 is a reducing transition grammar but is not (2-1) UI precedence grammar (note $ab \doteq c$ and $ab \lessdot c$), while G_2 is a (2-1) UI precedence grammar but is not a reducing transition grammar (both $(d, a, b) \to (1,D)$ and $(d,a,b) \to (1,E)$ are in the table). ∎

IV. CONCLUSIONS

It is interesting to notice that the class of reducing transition grammars was introduced more than 10 years ago and so, to the author's knowledge, is the first characterization given for the class of deterministic languages. Since this class does not contain other standard classes of grammars known to generate all deterministic languages, a new family was introduced. This family, the normal 2 form, weakly invertible, SMSP grammars is still sufficiently powerful to generate all deterministic languages and is included in the class of reducing transition grammars.

References

1. Eickel, J., Paul, M., Bauer, F. L., and Samelson, K., "A Syntax Controlled Generator of Formal Language Processors," Comm. ACM 6, 8 (August 1963), 451-455.

2. Wirth, N. and Weber, H., "EULER: A generalization of ALGOL, and its formal definition: Part I," Comm. ACM 9, 1 (January 1966), 13-23.

3. Morris, J. B., "A result on the relationship between Simple Precedence Languages and Reducing Transition Languages," Proceedings of the Symposium on Theory of Computing, ACM, (May 1970), 73-80.

4. Aho, A. V. and Ullman, J. D., The Theory of Parsing, Translation and Compiling, Prentice-Hall, Inc., 1972,3.

5. Ginsburg, S., The Mathematical Theory of Context-Free Languages, McGraw Hill, 1966.

6. Aho, A. V., Denning, P. J. and Ullman, J. D., "Weak and Mixed Strategy Precedence Parsing," JACM, 19, No. 2, (April 1972), 225-243.

7. Fisher, M. J., "Some properties of Precedence Languages," Proceedings of the Symposium on Theory of Computing, ACM, (May 1969), 181-190.