공학 연구논문집 제24권 제1호 pp. 191~204, 울산대학교
Journal of Engineering Research Vol. 24, No. 1, pp. 191~204, 1993.
University of Ulsan.

Transforming LR(*k*) Grammars into LALR(*k*) Grammars*

Myung-Joon Lee and Yang-Su Park
Department of Computer Science University of Ulsan

Abstract

We present a method of transforming an arbitrary LR(*k*) grammar G into an equivalent LALR(*k*) grammar by reducing the LR(*k*)-colored grammar for *G*. For this, we develop a few effective methods to reduce the LR(*k*)-colored grammar, preserving the SLR(*k*) and/or LALR(*k*) properties. In particular, an efficient algorithm for the case of k=1 is developed.

LR(k) 문법의 LALR(k) 문법으로의 변환에 관한 연구*

이명준 · 박양수
전자계산학과

요 약

본 연구에서는 임의의 LR(*k*) 문법을 그와 동일한 언어를 생성하는 LALR(*k*) 문법으로 변환하는 방법이 개발되었다. 이를 위하여 주어진 LR(*k*) 문법에 대한 LR(k)-colored 문법을 SLR(*k*) 성질 또는 LALR(*k*) 성질을 유지하면서 축약시키는 방법들이 연구되었으며, 특히 LR(1) 문법을 LALR(1) 문법으로 효율적으로 변환시키는 알고리즘이 개발되었다.

* This work was supported in part by NON-DIRECTED RESEARCH FUND, Korea Research Foundation, 1992, and by Korea Science & Engineering Foundation, Grant No. 901-11-1-002-2.

1. INTRODUCTION

Since the announcement of LR(k) grammars and their parsing [Knu65], much work has been done to discover subclasses of the LR(k) grammars for use in compilers. The efforts for discovering such subclasses bore some practical fruits such as well-known SLR(k) [Der71] and LALR(k) grammars [Der69, PCC85]. An SLR(k) grammar is an LR(k) grammar for which parsing conflicts in its LR(0) machine can be resolved by adding to every item the $FOLLOW_k$ set of the left part of the production in the item as lookahead information, whereas LALR(k) grammar is also an LR(k) grammar for which parsing conflicts in its LR(0) machine can be resolved by adding the state-dependent $FOLLOW_k$ set of the left part of the production in an item at each state. The class of LR(k) grammars properly includes the LALR(k) grammars, and the class of LALR(k) grammars properly includes the SLR(k) grammars. In particular, LALR(1) grammars have been used widespreadly for specifying the syntax of programming languages [Joh75, F&L88]

A context-free grammar G' is said to cover a context-free grammar G if both grammars generate the same language and there is a homomorphism from the parses of G' to those of G. Grammartical covers were introduced by Gray and Harrison [G&H72], and numerous covering results have been reported in the literature [Nij77, G&H72, R&H85]. Among others some results concerned with the deterministic parsing are notable since they enable us to transform grammars hard to parse into grammars easy to parse. They are also well surveyed in [Nij80, R&H85]. Recently the *LR(k)-colored grammar* was used for exploiting the covering result that every LR(k) grammar can be covered by an SLR(k) grammar[L&C91]. In this paper, we are concerned with the problem of transforming LR(k) grammars into equivalent LALR(k) grammars of which sizes are smaller than those of the associated LR(k)-colored grammars. For this, we present a few effective methods to reduce the LR(k)-colored grammar, preserving the SLR(k) and/or LALR(k) properties. Furthermore, an efficient algorithm for the case of k=1 is developed.

The organization of this paper is as follows. In Section 2, the notions of LR(k), SLR(k), LALR(k) grammars, LR(k)-colored grammars and grammatical covers are revisited. In Section 3, a few methods for reducing the LR(k)-colored grammar are developed which preserve the SLR(k) and/or LALR(k) property. In Section 4, the case of k=1 is handeled and an efficient algorithm for LR(1)-to-LALR(1) grammar transformation is presented. Finally, the results of the paper are summarized in Section 5.

2. Preliminaries

The section reviews some basic concepts concerning context-free grammars, LR(k) parsing, and grammatical coverings. For general background the reader is referred to [A&U73] and [L&C91].

The canonical collection of sets of LR(k)

items for G, denoted C_k, is defined recursively

$C_k =_s \{q_0\} \cup \{\mathrm{GOTO}(q,X) \mid q \in C_k, X \in V\}$,

where $q_0 = closure(\{[S' \to .S,\$]\})$, and

$\mathrm{GOTO}(q,X) = closure(\{[A \to \alpha X.\beta,u] \mid [A \to \alpha.X\beta,u] \in q\})$.

An element of C_k is said to be an LR(k) *state* over G. We call the state q_0 the *initial state*. The domain of the *GOTO* function is extended to $C_k \times V^*$ as follows:

$GOTO(q,\varepsilon)=q$ and $GOTO(q,X\gamma)=GOTO(GOTO(q,X),\gamma)$.

A generalized LR(k) parser for the whole class of context-free grammars is defined by the following formal system called *LR(k) machine*.

Definition 2.1. (*LR(k) machine*) The *LR(k) machine* for G is a 4-tuple $LRM_k(G)=(C_k, GOTO, ACTION, q_0)$, where C_k, *GOTO*, and q_0 are as was stated in the formal section; *ACTION* is a function from $C_k \times FIRST_k(\Sigma^*\$)$ to subsets of $\{shift, accept\} \cup \{reduce\ \pi \mid \pi \in P\}$ defined by

$ACTION(q,u)=\{shift\}$ if $[A \to \alpha.\beta,v] \in q, A \neq S', \beta \neq \varepsilon$, and $u \in EFF_k(\beta v)$

$\cup \{reduce\ \pi\}$ if $\pi = A \to \alpha$, $[A \to \alpha.,u] \in q$, and $A \neq S'$

$\cup \{accept\}$ if $[S' \to S.,\$] \in q$, and $u=\$$,

where EFF_k, the ε-free $FIRST_k$ function, is

$EFF_k(\alpha)=\{PREF_k(\alpha) \mid \alpha \in \Sigma^*\$\}$

$\cup \{PREF_k(x) \mid \alpha \overset{*}{\Rightarrow}_{rm} \beta \Rightarrow_{rm} x,$

$\beta \neq Ax$ for *all* $A \in N, x \in \Sigma^*\$\}$.

It would be noted that if $ACTION(q,u)$ contains *shift*, then the state q contains an LR(k) item $[A \to \alpha.a\beta,v]$ with $u \in FIRST_k(a\beta v)$ because of the definition of the *closure* function and the EFF_k function above.

Definition 2.2. (*right-cover*) Let $G_1=(N_1, \Sigma, P_1, S_1)$ and $G_2=(N_2, \Sigma, P_2, S_2)$ be CFG' s. G_2 is said to be a right-cover of G_1 if there is a homomorphism h from P_2 to P_1 such that

(1) if there is a rightmost derivation in G_1 such that $S_1 \overset{*}{\Rightarrow}_{\pi_1\cdots\pi_n} x$, then there is a rightmost derivation in G_2 such that $S_2 \overset{*}{\Rightarrow}_{\pi'_1\cdots\pi'_m} x$, and $h(\pi'_1\cdots\pi'_m)=\pi_1\cdots\pi_n$;

(2) if there is a rightmost derivation in G_2 such that $S_2 \overset{*}{\Rightarrow}_{\pi'_1\cdots\pi'_m} x$, then there is a rightmost derivation in G_1 such that $S_1 \overset{*}{\Rightarrow}_{h(\pi'_1\ldots\pi'_m)} x$.

Obviously $L(G_1)=L(G_2)$ if G_2 is a right-cover of G_1.

Definition 2.3. (*SLR(k)-cover*) Let $G_1=(N_1, \Sigma, P_1, S_1)$ and $G_2=(N_2, \Sigma, P_2, S_2)$ be CFG' s. G_2 is said to be an *SLR(k)-cover* of G_1 if G_2 is a right-cover of G_1 and G_2 is SLR(k).

Definition 2.4. (LALR*(k)-cover*) Let $G_1=(N_1, \Sigma, P_1, S_1)$ and $G_2=(N_2, \Sigma, P_2, S_2)$ be CFG s. G_2 is said to be an SLR(k)-cover of G_1 if G_2 is a right-cover of G_1 and G_2 is LALR(k).

Definition 2.5. (*LR(k)-colored grammar [L&C91]*) Let a CFG $G=(N, \Sigma, P, S)$, and $LRM_k(G)=(C_k, GOTO, ACTION, q_0)$. The *LR(k)-colored grammar* for G is $\mathbf{G}=(\mathbf{N}, \Sigma, \mathbf{P}, \mathbf{S})$, where

(1) $\mathbf{N}=\{\mathbf{S}\} \cup \{X^q \mid q \in C_k, [A \to \alpha X\beta,u] \in q, X \in V\}$

$\cup \{\pi^q \mid q \in C_k, [A \to \alpha., u] \in q, A \neq S', \pi$ *is* $A \to \alpha \in P\}$,

The set of new vocabularies, $N \cup \Sigma$, is denoted by V. For notational convenience, we classify N into four disjoint sets, $\{S\}$, N_N, N_Σ, and N_P, as follows:

$N_N = \{A^q \mid A \in N, A^q \in N\}$, $N_\Sigma = \{a^q \mid a \in \Sigma, a^q \in N\}$, $N_P = \{\pi^q \mid \pi \in P, \pi^q \in N\}$; and the set $N_N \cup N_\Sigma$ is denoted by N_V.

(2) $P = \{S \to S^{q_0})$
$\cup \bigcup_{q \in C_k} \{A^q \to \theta(q,\alpha) \cdot \pi^{q'} \mid A \neq S' [A \to .\alpha, u] \in q, \pi$ is $A \to \alpha, q' = GOTO(q,\alpha)\}$
$\cup \bigcup_{a^q \in N_\Sigma} \{a^q \to a\} \cup \bigcup_{\pi^q \in N_p} \{\pi^q \to \varepsilon\}$,

where $\theta(q,\alpha)$ is a function from $C_k \times V^*$ to N_v^* defined by

$\theta(q,\varepsilon) = \varepsilon$ and $\theta(q, X\alpha) = X^q \cdot \theta(GOTO(q, X),\alpha)$
if X^q is in N_V (or equivalently $GOTO(q, X)$ is in C_k).

Theorem 2.6. ([L&C91]) *G is an SLR(k)-cover for G if G is LR(k).*

3. Transformation Preserving LALR(k) Property

For LR(k) grammar G, the LR(k)-colored grammar for G is also an LALR(k)-cover of G because the LR(k)-colored grammar is SLR(k) and thus also LALR(k). However, since the LR(k)-colored grammar is often of somewhat unacceptable size, the chapter presents some useful methods to reduce the LR(k)-colored grammar with the LALR(k) covering property preserved.

3. 1 Basic Reduction of LR(k)-colored Grammar

According to Theorem 2.6, we know that **G** is an SLR(k)-cover for G. But as far as SLR(k) covering is concerned, **G** has some kinds of unnecessary nonterminals. Hence we introduce a grammar which is a natually reduced version of **G** but preserves the SLR (k) covering property.

Definition 3.1. We define a fine homomorphism h: $V \to \{S\} \cup N_N \cup \Sigma \cup \{\varepsilon\}$ by

$$h(X) = \begin{cases} \varepsilon, \text{ if } X = \pi^q \in N_p \\ a, \text{ if } X = a^q \in N_\Sigma \\ X, \text{ if } X \in \{S\} \cup N_N \cup \Sigma. \end{cases}$$

Difinition 3.2. Let $G=(N, \Sigma, P, S)$ be a CFG and $\mathbf{G}=(\mathbf{N}, \Sigma, \mathbf{P}, S)$ be the LR(k)-colored grammar for G. The *reduced LR(k)-colored grammar* for G is $\mathbf{G}_r=(\mathbf{N}_r, \Sigma, \mathbf{P}_r, S)$, where

(1) $N_r = \{S\} \cup N_N$,

(2) $\mathbf{P}_r = \{h(A \to \alpha) \mid A \to \alpha \in \mathbf{P}\} - \{\varepsilon\}$, where

$$h(A \to \alpha) \begin{cases} h(A) \to h(a), \text{ if } A \in N_r \\ \varepsilon, \quad \text{if } A \in N_P \cup N_\Sigma. \end{cases}$$

Example 3.3 Let G be CFG $G=(\{S, A, B\}, \{a, b, c, d, e, \$\}, P, S)$, where P is composed of the following productions:

π_1: $S \to aAdS$, π_2: $S \to \varepsilon$, π_3: $S \to bBd$, π_4: $S \to aBe$,

π_5: $S \to bAe$, π_6: $A \to c$, π_7: $B \to c$.

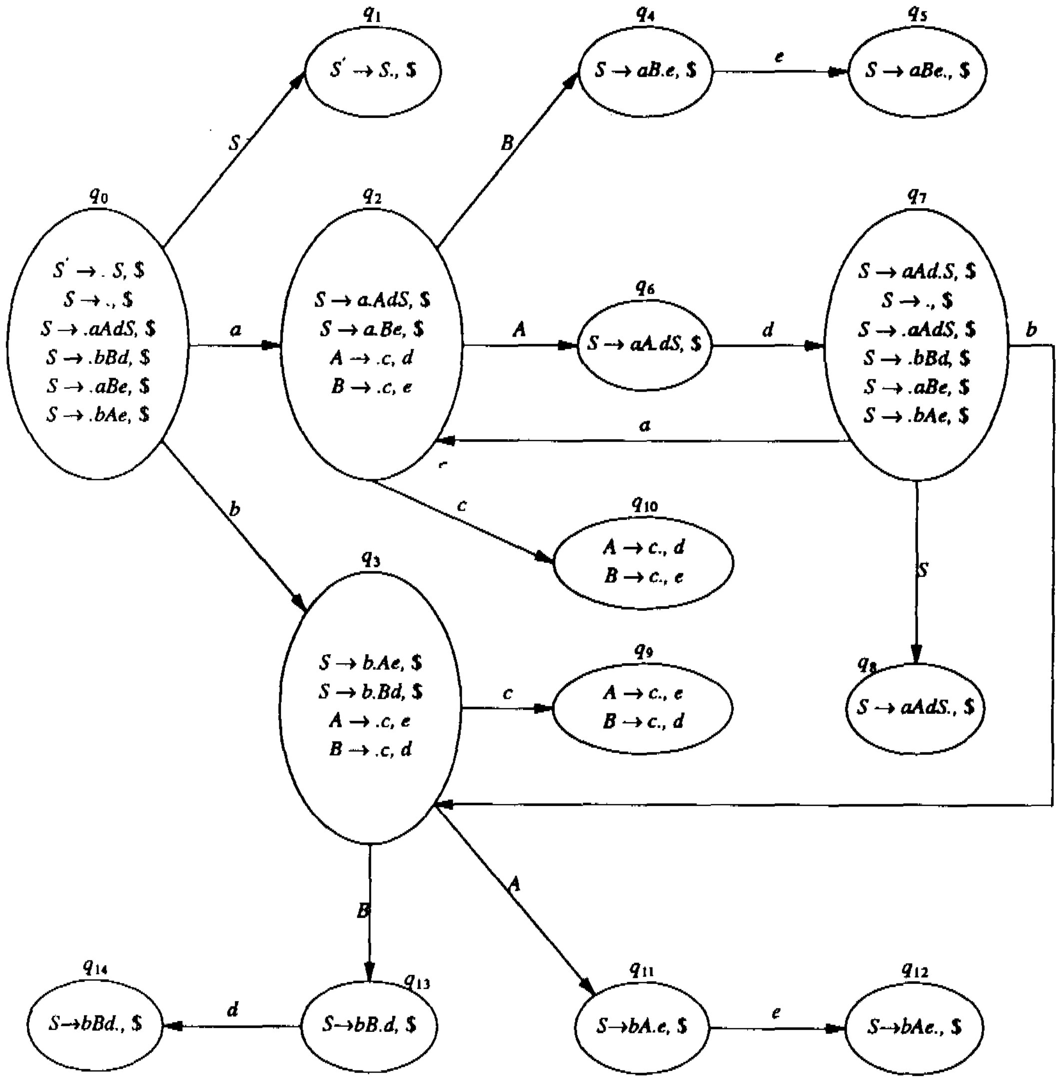

Figure 3.1 LR(1) automation for G

The LR(1) automation for G is presented in Figure 3.1. By Definition 2.5, the LR(k)-colored grammar for G is $\boldsymbol{G}=(N,\Sigma, \boldsymbol{P}, S)$, where

(1) $N = \{S\} \cup N_N \cup N_\Sigma \cup N_P$, where

$N_N = \{S^{q_0}, S^{q_7}, A^{q_2}, A^{q_3}, B^{q_2}, B^{q_3}\}$

$N_\Sigma = \{a^{q_0}, a^{q_7}, b^{q_0}, b^{q_7}, c^{q_2}, c^{q_3}, d^{q_6}, d^{q_{13}}, e^{q_4}, e^{q_{11}}\}$

$N_P = \{\pi_1^{q_8}, \pi_2^{q_0}, \pi_2^{q_7}, \pi_3^{q_{14}}, \pi_4^{q_5}, \pi_5^{q_{12}}, \pi_6^{q_9}, \pi_6^{q_{10}}, \pi_7^{q_9}, \pi_7^{q_{10}}\}$;

(2) $\boldsymbol{P}$ is composed of the following productions:

$S \to S^{q_0}$,

$S^{q_0} \to \pi_1^{q_0} \mid a^{q_0} A^{q_2} d^{q_6} S^{q_7} \pi_1^{q_8} \mid a^{q_0} B^{q_2} e^{q_4} \pi_4^{q_5} \mid b^{q_0} B^{q_3} d^{q_{13}} \pi_3^{q_{14}} \mid b^{q_0} A^{q_3} e^{q_{11}} \pi_5^{q_{12}}$,

$S^{q_7} \to \pi_2^{q_7} \mid a^{q_7} A^{q_2} d^{q_6} S^{q_7} \pi_1^{q_8} \mid a^{q_7} B^{q_2} e^{q_4} \pi_4^{q_5} \mid b^{q_7} B^{q_3} d^{q_{13}} \pi_3^{q_{14}} \mid b^{q_7} A^{q_3} e^{q_{11}} \pi_5^{q_{12}}$,

$A^{q_2} \to c^{q_2} \pi_6^{q_{10}}$, $A^{q_3} \to c^{q_3} \pi_6^{q_9}$, $B^{q_2} \to c^{q_2} \pi_7^{q_{10}}$, $B^{q_3} \to c^{q_3} \pi_7^{q_9}$,

$X^q \to X$ for all $X^q \in N_\Sigma$ $\quad \pi^q \to \varepsilon$ for all $\pi^q \in N_P$.

Now by Definition 3.2, the reduced LR(k)-colored grammar for G is $\boldsymbol{G}_r=(N_r, \Sigma, \boldsymbol{P}_r, S)$, where

(1) $N_r=\{S\} \cup N_N = \{S, S^{q_0}, S^{q_7}, A^{q_2}, A^{q_3}, B^{q_2}, B^{q_3}\}$,

(2) $\boldsymbol{P}_r$ is composed of the following productions:

$S \to S^{q_0}$,

$S^{q_0} \to aA^{q_2}dS^{q_7} \mid aB^{q_2}e \mid bB^{q_3}d \mid bA^{q_3}e \mid \varepsilon$,

$S^{q_7} \to aA^{q_2}dS^{q_7} \mid aB^{q_2}e \mid bB^{q_3}d \mid bA^{q_3}e \mid \varepsilon$,

$A^{q_0} \to c$, $A^{q_3} \to c$, $B^{q_2} \to c$, $B^{q_3} \to c$.

Lemma 3.4. $\boldsymbol{G}_r$ *is SLR(k) if and only if* $\boldsymbol{G}$ *is SLR(k).*

Proof. (*If*): Suppose that $\boldsymbol{G}$ is SLR(k). Let us consider the following two cases.

(1) (reduce-reduce conflicts) Suppose that $[A^p \to \alpha_r.]$ and $[B^s \to \beta_r.]$ are distinct items contained in an arbitrary LR(0) state over $\boldsymbol{G}_r$. Let $\alpha=\theta(p, \alpha)$, $\beta=\theta(p, \beta)$, $\alpha_r=h(\alpha)$, and $\beta_r=h(\beta)$; and let q be the state GOTO(p, α) (=GOTO(s, β_r). Then the items $[A^p \to \alpha.\pi_1^q]$ and $[B^s \to \beta.\pi_2^q]$ are contained in the same LR(0) state over $\boldsymbol{G}$, and thus the items $[\pi_1^q \to .]$ and $[\pi_2^q \to .]$ are also. Since $\boldsymbol{G}$ is SLR(k), the following equation holds:

$$FOLLOW_k^{G}(\pi_1^q) \cap FOLLOW_k^{G}(\pi_2^q)=\varnothing.$$

Then the following equation also holds:

$$FOLLOW_k^{G_r}(A^p) \cap FOLLOW_k^{G_r}(B^s)=\varnothing,$$

because, $FOLLOW_k^G(\pi_1^q)$ includes $FOLLOW_k^{G_r}(A^p)$ ($=FOLLOW_k^G(A^p)$) and $FOLLOW_k^G(\pi_2^q)$ includes $FOLLOW_k^{G_r}(B^s)$ ($=FOLLOW_k^G(B^s)$).

(2)(shift-reduce conflicts) Suppose that $[A^p \rightarrow \alpha_r . X_r\beta_r]$ and $[B^s \rightarrow \gamma_r .]$ are distinct items contained in an arbitary LR(0) state over G_r. Let $\alpha = \theta(P, \alpha)$, $q = GOTO(P, \alpha)$, $X = X^q$, $\beta = \theta(GOTO(q, x), \beta)$, $\alpha_r = h(\alpha)$, $\beta_r = h(\beta)$, $\gamma_r = h(\gamma)$, and $X_r = h(X)$. Then the items $[A^p \rightarrow \alpha . X^q\beta\pi_1^t]$, $[B^s \rightarrow \gamma . \pi_2^q]$ and $[\pi_2^q \rightarrow .]$ are contained in the same LR(0) state over G, where $\pi_1 = A \rightarrow \alpha X\beta$, $\pi_2 = B \rightarrow \gamma$, and $t = GOTO(P, \alpha X\beta)$. Since G is SLR(K), the following equation holds:

$$EFF_k^G(X^q\beta\pi_1^t) \oplus_k FOLLOW_k^G(A^p) \cap FOLLOW_k^G(\pi_2^q) = \emptyset.$$

Then the following equation also holds:

$$EFF_k^{G_r}(X_r\beta_r) \oplus_k FOLLOW_k^{G_r}(A^p) \cap FOLLOW_k^{G_r}(B^s) = \emptyset,$$

because

$$EFF_k^{G_r}(X_r\beta_r) \oplus_k FOLLOW_k^{G_r}(A^p) = EFF_k^G(X^q\beta\pi_1^s) \oplus_k FOLLOW_k^G(A^p),$$

and $FOLLOW_k^G(\pi_2^q)$ includes $FOLLOW_k^{G_r}(B^s)$
Now, by (1) and (2), G_r is SLR(k)

(*Only If*): Clear. □

Theorem 3.5. *G_r is an SLR(k)-cover for G if G is LR(k).*

Proof. First, we define a very fine homomorphism h_r from the vocabulary of G_r to that of G by

$$h_r(X) = \begin{cases} S, \text{ if } X=S \\ a, \text{ if } X=a\in\Sigma \\ X, \text{ if } X=X^q \in (N_r - \{S\}). \end{cases}$$

Then the following property obviously holds.

Property-1. *Let p be $GOTO(q_0, \gamma)$, and γ be $h(\theta(q_0, \gamma))$, where h is defined in Definition 3.1. Then G permits a derivation $S \Rightarrow_{rm}^* \gamma Ax$ if and only if G_r permits derivation $S \Rightarrow_{rm}^* \gamma A^p x$.*

Now let us extend the homomorphism h_r to a homomorphism between productions by the following. The homomorphism h_r: $P_r \rightarrow P$ is defined by

$$h_r(A \rightarrow \alpha) \begin{cases} \varepsilon, \text{ if } A=S \\ h_r(A) \rightarrow h_r(\alpha), \text{ otherwise.} \end{cases}$$

Then Propetry-1 and the extended homomorphism h_r simply yield the following property which says that G_r right covers G under the homomorphism h_r.

Property-2. *There is a rightmost derivation in G_r such that $S \Rightarrow_{\pi_1 \cdots \pi_m}^* x$ and $h_r(\pi_1 \cdots \pi_m) = \pi_1 \cdots \pi_n$ if and only if there is a rightmost derivation in G such that $S \Rightarrow_{\pi_1 \cdots \pi_n}^* x$.*

From Lemma 3.4 and Property-2, the theorem holds. □

3.2 First Level Reduction

To reduce an LR(k)-colored grammar, the previous section uses the essential feature of the LR(k)-colored grammar. This section utilizes the information which can be gathered from SLR(k) parsing conflicts dependent on a given context-free grammar. We begin by introducing a new notion of *SLR(k)-conflictable nonterminals*, which correspond to parsing actions resulting in SLR(k) parsing conflicts.

Definition 3.6. (*SLR(k)-conflictable*) We say that a nonterminal A is *SLR(k)-conflictable in p* over G if an LR(0) item of the form $[A \rightarrow \alpha.\beta]$ exhibits a SLR(k) parsing conflict in state p over the LR(0) machine for G.

For capturing the affectedness of SLR(k)-conflictable nonterminals, we introduce a relation between nonterminals.

Definition 3.7. (ends)

A *ends* B if $A \rightarrow \alpha B \beta \in P$ and $\beta \Rightarrow^* \varepsilon$

To obtain the LR(k) states which have the same core as the core of a given LR(0) state, a funtion is defined:

Definition 3.8. (*LR(k)-color*) We define a funtion *LR(k)-color*: $C_0 \rightarrow 2^{C_k}$:

$LR(k)\text{-}color(p)=\{q \mid p=Core(q)\}$

The nonterminals which affect directly and/or indirectly the FOLLOW sets of SLR (k)-conflictable nonterminals can be captured as follows:

Definition 3.9. (*SLR(k)-sensitive*) We say that a nonterminal $A^q \in N$ is *SLR(k)-sensitive* if A is SLR(k)-conflictable in p and $q \in LR(k)\text{-}color(p)$.

Now the nonterminals which do not affect the FOLLOW set of SLR(k)-conflictable nonterminals neither directly nor indirectly can be captured as follows:

Definition 3.10. (*SLR(k)-free*) We say that a nonterminal $A^q \in N$ is *SLR(k)-free* if the relation A^q *ends** B^r does not hold for any SLR (k)-sensitive nonterminal B^r.

By utilizing the SLR(k)-free nonterminals, a new grammar can be obtained, which rightly covers the reduced LR(k)-colored grammar, according to the following definitions.

Definition 3.11. (*vocabulary homomorphism*) We define a very fine homomorphism h_0 between the vocabularies of $\boldsymbol{G_r}$

$$h_0(X)= \begin{cases} S, \text{ if } X=S \\ \text{A, if } X=A^q \text{ is } SLR(k)\text{-}free \\ X, \text{ otherwise.} \end{cases}$$

Definition 3.12. (*first-level reduction*) Let $G=(N, \Sigma, P, S)$ be a CFG and $\boldsymbol{G_r}=(N_r, \Sigma, \boldsymbol{P_r}, S)$ be the *reduced LR(k)-colored grammar* for G. The *first-level reduced LR(k)-colored grammar* for G is $\boldsymbol{G_0}=(N_0, \Sigma, \boldsymbol{P_0}, S)$, where

$\boldsymbol{P_0}=\{h_0(A) \rightarrow h_0(\alpha) \mid A \rightarrow \alpha \in \boldsymbol{P_r}\}$.

The above definitions naturally lead to the following lemma.

Lemma. 3.13. *If* $\boldsymbol{G_r}$ *is SLR(k), then* $\boldsymbol{G_0}$ *is SLR(k).*

For an LR(k) grammar G, the grammar $\boldsymbol{G_0}$ preserves the SLR(k) covering property of the reduced LR(k)-colored grammar $\boldsymbol{G_r}$ as will be shown in the following theorem.

Theorem. 3.14. *If G is LR(k), then* $\boldsymbol{G_0}$ *is an SLR(k)-cover for G.*

Proof. First, we define a very fine homomorphism h_0 from the vocabulary of $\boldsymbol{G_0}$ to that of G by

$$h_0(X)= \begin{cases} S, \text{ if } X=S \\ a, \text{ if } X=a \in \Sigma \\ X, \text{ if } X=X^r \in (N_0 - \{S\}). \end{cases}$$

The homomorphism h_0 is extended to a homomorphism between productions by the following. The homomorphism $h_0: P_0 \to P$ is

$$h_0(A \to \alpha) \begin{cases} \varepsilon, & \text{if } A=S \\ h_0(A) \to h_0(\alpha), & \text{otherwise.} \end{cases}$$

defined by

Now G_0 is a right cover for G under the homomorphism h_0; and thus, according to Lemma 3.13, the theorem holds. □

3.3 Second Level Reduction

To reduce the first-level reduced LR(k)-colored grammar, this section utilizes the information which can be gathered from LALR(k) parsing conflicts dependent on a given context-free grammar. We begin by introducing another notion of *LALR(k)-conflictable nonterminals*, which correspond to parsing actions resulting in LALR(k) parsing conflicts.

Definition 3.15. (*LALR(k)-conflictable*) We say that a nonterminal A is *LALR(k)-conflictable in p* over G if an LR(0) item of the form $[A \to \alpha.\beta]$ exhibits a LALR(k) parsing conflict in state p over the LR(0) machine for G.

The nonterminals which affect directly and/or indirectly the FOLLOW sets of LALR (k)-conflictable nonterminals can be captured as follows:

Definition 3.16. (*LALR(k)-sensitive*) We say that a nonterminal $A^q \in N$ is *LALR(k)-sensitive* if A is LALR(k)-conflictable in p and $q \in LR(k)\text{-}color(p)$.

Now the nonterminals which do not affect the FOLLOW set of LALR(k)-conflictable nonterminals neither directly nor indirectly can be captured as follows:

Definition 3.17. (*LALR(k)-free*) We say that a nonterminal $A^q \in N$ is *LALR(k)-free* if the relation A^q *ends** B^r does not hold for any LALR(k)-sensitive nonterminal B^r.

By utilizing the LALR(k)-free nonterminals, another grammar can be obtained, which rightly covers the first-level reduced LR(k)-colored grammar, according to the following definitions.

Definition 3.18. (*vocabulary homomorphism*) We define a very fine homomorphism h_l between the vocabularies of G_0

$$h_l(X) = \begin{cases} S, & \text{if } X=S \\ A, & \text{if } X=A^q \textit{ is LALR(k)-free} \\ X, & \text{otherwise.} \end{cases}$$

Definition 3. 19. (*second-level reduction*) Let $G=(N, \Sigma, P, S)$ be a CFG and $G_0=(N_0, \Sigma, P_0, S)$ be the *first-level reduced LR(k)-colored grammar for G*. The *second-level reduced LR (k)-colored grammar* for G is $G_l=(N_l, \Sigma, P_l, S)$, where

$P_l=\{h_l(A) \to h_l(\alpha) \mid A \to \alpha \in P_0\}$.

The above definitions naturally lead to the following lemma.

Lemma. 3.20. *If G_0 is SLR(k), then G_l is LALR(k).*

For an LR(k) grammar G, the grammar G_l preserves the LALR(k) covering property of the first-level reduced LR(k)-colored grammar G_0 as will be shown in the following theorem.

Theorem. 3.21. *If G is LR(k), then G_l is an LALR(k)-cover for G under the homomorphism h_0.*

Proof. Fist, we define a very fine homomorphism h_l from the vocabulary of G_l to that of G by

$$h_l(X)= \begin{cases} S, \text{ if } X=S \\ a, \text{ if } X=a\in\Sigma \\ X, \text{ if } X=X^a\in(N_l - \{S\}). \end{cases}$$

The homomorphism h_l is extended to a homomorphism between productions by the following. The homomorphism h_l: $P_l \rightarrow P$ is defined by

$$h_l(A \rightarrow \alpha) \begin{cases} \varepsilon, \text{ if } A=S \\ h_l(A)\rightarrow h_l(\alpha), \text{ otherwise.} \end{cases}$$

Now G_l is a right cover for G under the homomorphism h_l; and thus, according to Lemma 3.20, the theorem holds. □

4. LR(1)-to-LALR(1) grammar transformation

In general, an LR(k) grammar is not LALR (k), the LALR(k) parser for the LR(k) grammar can have some inconsitent states which exhibit shift-reduce parsing conflicts [Hei81] as well as reduce-reduce parsing conflicts. In the case of k=1, however, any inconsistent states of the LALR(1) parser for an LR(1) grammar inherently does not contain shift-reduce parsing conflicts [A&U73]. Thus LR(1) grammars can be transformed into LALR(1) form more effectively than LR($k\geq$ 2) grammars can. The following definition captures this fact as a modified version of Definition 3.15.

Definition 4.1. (*LALR(k)-conflictable*) We say that a nonterminal A is *LALR(1)-conflictable in p* over G if an LR(0) item of the form $[A\rightarrow\alpha.]$ exhibits an LALR(1) parsing conflict in state p over the LR(0) machine for G.

Following the discussion of the previous sections, and arbitrary LR(1) grammar can be effectively transformed into an LALR(1) grammar as stated in Algorithm 4.2. Notice that Algorithm 4.2 does not compute the first-level reduced LR(1)-colored grammar since every LALR(1)-conflictable symbol is a SLR(1)-conflictable symbol and thus every SLR(1)-free symbol is an LALR(1)-free symbol.

Algorithm 4.2. (*LR(1)-to-LALR(1) transformation*)

step-1.
(*Get LR(1)-colored grammar*) Compute the LR(1)-colored grammar G for an LR(1) grammar G according to Definition 2.5.

step-2.
(*Get reduced LR(1)-colored grammar*) Compute G_r, the reduced LR(1)-colored grammar, from G according to Definition 3.2.

step-3.
(*Get LALR(1)-conflictable symbols*) Compute LALR(1)-conflictable nonterminals over G_r, which describe LALR (1) parsing conflicts over the LR(0) machine for G according to Definition

4.1.

step-4.

(*Get LALR(1)-conflict-sensitive symbols*) Compute LALR(1)-sensitive nonterminals which directly affect LALR(1) parsing conflicts by analyzing LALR(1)-conflictable nonterminals, according to Definition 3.16.

step-5.

(*Get LALR(1)-conflict-free symbols*) Compute LALR(1)-free nonterminals over $\boldsymbol{G_r}$, which affect any LALR(1)-sensitive nonterminals neither directly nor indirectly, according to Definition 3.17.

step-6.

(*Get LALR(1) grammar*) Compute an LALR(1) grammar equivalent to LR(1) grammar G, $\boldsymbol{G_l}$, the second-level-reduced LR(1)-colored grammar, according to Definition 3.19.

Algorithm 4.2 transforms LR(1) grammars into a equivalent LALR(1) grammars as the following example shows.

Example 4.3. Let G be a CFG $G=(\{S, A, B\}, \{a, b, c, d, e, \$\}, P, S)$, where P is composed of the following productions:

$\pi_1 : S \rightarrow aAdS$, $\pi_2 : S \rightarrow \varepsilon$, $\pi_3 : S \rightarrow bBd$, $\pi_4 : S \rightarrow aBe$,

$\pi_5 : S \rightarrow bAe$, $\pi_6 : A \rightarrow c$, $\pi_7 : B \rightarrow c$.

The LR(1) automation for G is presented in Figure 3.1.

step-1. (*Get LR(1)-colored grammar*)

The LR(1)-colored grammar for G is $\boldsymbol{G}=(\boldsymbol{N}, \Sigma, \boldsymbol{P}, \boldsymbol{S})$, where

(1) $\boldsymbol{N} = \{S\} \cup \boldsymbol{N_N} \cup \boldsymbol{N_\Sigma} \cup \boldsymbol{N_P}$, where

$\boldsymbol{N_N} = \{S^{q_0}, S^{q_7}, A^{q_2}, A^{q_3}, B^{q_2}, B^{q_3}\}$

$\boldsymbol{N_\Sigma} = \{a^{q_0}, a^{q_7}, b^{q_0}, b^{q_7}, c^{q_2}, c^{q_3}, d^{q_6}, d^{q_{13}}, e^{q_4}, e^{q_{11}}\}$

$\boldsymbol{N_P} = \{ \pi_1^{q_8}, \pi_2^{q_0}, \pi_2^{q_7}, \pi_3^{q_{14}}, \pi_4^{q_5}, \pi_4^{q_5}, \pi_5^{q_{12}}, \pi_6^{q_9}, \pi_6^{q_{10}}, \pi_7^{q_9}, \pi_7^{q_{10}} \}$;

(2) $\boldsymbol{P}$ is composed of the following productions:

$S \rightarrow S^{q_0}$,

$S^{q_0} \rightarrow \pi_2^{q_0} \mid a^{q_0} A^{q_2} d^{q_6} S^{q_7} \pi_1^{q_8} \mid a^{q_0} B^{q_2} e^{q_4} \pi_4^{q_5}, \mid b^{q_0} B^{q_3} d^{q_{13}} \pi_3^{q_{14}} \mid b^{q_0} A^{q_3} e^{q_{11}} \pi_5^{q_{12}}$,

$S^{q_7} \rightarrow \pi_2^{q_7} \mid a^{q_7} A^{q_2} d^{q_6} S^{q_7} \pi_1^{q_8} \mid a^{q_7} B^{q_2} e^{q_4} \pi_4^{q_5}, \mid b^{q_7} B^{q_3} d^{q_{13}} \pi_3^{q_{14}} \mid b^{q_7} A^{q_3} e^{q_{11}} \pi_5^{q_{12}}$,

$A^{q_2} \rightarrow c^{q_2} \pi_6^{q_{10}}$, $A^{q_3} \rightarrow c^{q_3} \pi_6^{q_9}$, $B^{q_2} \rightarrow c^{q_2} \pi_7^{q_{10}}$, $B^{q_3} \rightarrow c^{q_3} \pi_7^{q_9}$.

$X^q \rightarrow X$ for all $X^q \in \boldsymbol{N_\Sigma}$ $\pi^q \rightarrow \varepsilon$ for all $\pi^q \in \boldsymbol{N_P}$

step-2. (*Get reduced LR(1)-colored grammar*)

The reduced LR(1)-colored grammar for G is $\boldsymbol{G_r}=(\boldsymbol{N_r}, \Sigma, \boldsymbol{P_r}, \mathrm{S})$, where

(1) $N_r = \{S\} \cup N_N = \{S, S^{q_0}, S^{q_7}, A^{q_2}, A^{q_3}, B^{q_2}, B^{q_3}\}$,

(2) P_r is composed of the following productions:

$S \to S^{q_0}$,

$S^{q_0} \to aA^{q_2}dS^{q_7} \mid aB^{q_2}e \mid bB^{q_3}d \mid bA^{q_3}e \mid \varepsilon$,

$S^{q_7} \to aA^{q_2}dS^{q_7} \mid aB^{q_2}e \mid bB^{q_3}d \mid bA^{q_3}e \mid \varepsilon$,

$A^{q_2} \to c, \quad A^{q_3} \to c, \quad B^{q_2} \to c, \quad B^{q_3} \to c.$

step-3. *(Get LALR(1)-conflictable symbols)*

According to Definition 4.1, the set of LALR(1)-conflictable nonterminals in G_r are $\{A, B\}$.

step-4. *(Get LALR(1)-conflict-sensitive symbols)*

According to Definition 3.16, the set of LALR(1)-conflict-sensitive nonterminals in G_r is $\{A^{q_2}, A^{q_3}, B^{q_2}, B^{q_3}\}$.

step-5. *(Get LALR(1)-conflict-free symbols)*

According to Definition 3.17, the set of LALR(1)-conflict-free nonterminals in G_r is $\{S^{q_0}, S^{q_7}\}$.

step-6. *(Get LALR(1) grammar)*

According to Definition 3.19, the second-level-reduced LR(1)-colored grammar for G is $G_l = (N_l, \Sigma, P_l, S)$, where

(1) $N_l = \{S\} \cup N_N = \{S, S, A^{q_2}, A^{q_3}, B^{q_2}, B^{q_3}\}$,

(2) P_l is composed of the following productions:

$S \to S$,

$S \to aA^{q_2}dS \mid aB^{q_2}e \mid bB^{q_3}d \mid bA^{q_3}e \mid \varepsilon$,

$A^{q_2} \to c, \quad A^{q_3} \to c, \quad B^{q_2} \to c, \quad B^{q_3} \to c.$

If we eliminate the useless augmented start production $S \to S$ from G_l, the resulting LALR(1) grammar equivalent to G becomes

$S \to aA^{q_2}dS \mid aB^{q_2}e \mid bB^{q_3}d \mid bA^{q_3}e \mid \varepsilon$,

$A^{q_2} \to c, \quad A^{q_3} \to c, \quad B^{q_2} \to c, \quad B^{q_3} \to c.$

5. Conclusion

We have presented a method of transforming an LR(k) grammar G into an equivalent LALR(k) grammar by reducing the *LR(k)-colored grammar*, whose grammar symbols are appropriately colored by information on LR(k) states over G so that the actions of the

LR(k) parser for G may be transfigured into the grammar symbols.

For this, we have developed a few effective methods to reduce the LR(k)-colored grammar for a given LR(k) grammar, preserving the SLR(k) and/or LALR(k) properties, by utilizing the essential characteristics of the LR(k)-colored grammar and the information derived from SLR(k) and/or LALR(k) parsing conflicts. All the transformation methods have been designed under the notion of grammatical covering; the resulting grammars transformed by the methods rightly cover the original grammars, repectively. In particular, an efficient method is also developed which transforms every LR(1) grammar into an equivalent LALR(1) grammar.

REFERENCES

[A&U73] Aho, A. V., and Ullman, J. D., The theory of parsing, translation and compiling, Vol. 1-2. Englewood Cliffs, N.J.;Prentice-Hall 1972, 1973.

[Der69] DeRemer, F.L., Practical Translator for LR(k) Languages, Ph.D. thesis, Dept. of Electrical Engineering, MIT, Cambridge, Mass., 1969.

[Der71] DeRemer, F.L., Simple LR(k) grammars, *CACM* **14**(1971), 453-460.

[F&L88] Fischer, C. N. and LeBlanc, R. J., Crafting a compiler, The Benjamin/ Cummings Publishing Company, Inc., 1988.

[G&H72] Gray, J.N., and Harrison, M. A., On the covering and reduction problems for context-free grammars, *J. ACM* **12** (1972), 675-698.

[Joh75] Johnson, S.C., Yacc: Yet another compiler, Computing Science Technical Report No. 32(1975), AT&T Bell Laboratories, Murray Hill, N. J.

[Knu65] Knuth, D. E., On the translation of languages from left to right, *Inf. Control* **8**(1965), 607-639.

[L&C91] Lee, M. J. and Choe, K. M., SLR (k) covering for LR(k) grammars, *Informa-tion Processing Letters*, **37** (1991), 337-347.

[Nij77] Nijholt, A., On the covering of parsable grammars, *J. Comput. System Sci.* **15**(1977), 99-110.

[Nij80] Nijholt, A., Context-Free Grammars: Covers, Normal Forms, and Parsing, Lect. Notes in Comp. Sci. **93**(1980), Springer, Berlin.

[PCC85] Park, J. C. H., Choe, K. M., and Chang, C. H., A new analysis of LALR for malisms, *ACM Trans. Program Lang. Syst.* **7**(1985), 159-175.

[R&H85] Rosenkrantz, D. J., and Hunt, III, H. B., Testing for grammartical coverings, *Theor. Comput. Sci.* **38**(1985), 323-341. *ACM Trans. Program Lang. Syst.* **9** (1987), 543-566.