7. Construction and Implementation of LR(1) Parsers

Implementation methods of LR(1) parsers:

table-driven,

set of program statements of rules

7.1 Construction of SLR(1) Parsers

To construct SLR(1) parser:

construct canonical LR(0) collection and deterministic (canonical) LR(0) machine,

simply add the lookahead symbols to the rules of the LR(0) parser.

In deterministic LR(0) machine for G;

$Valid_0(\varepsilon) = {}^{1}{}^{*}\{[S \to \cdot\omega] \mid S \to \omega \in P\}$,

$Goto(Valid_0(\gamma), X) = Valid_0(\gamma X)$.

Let $G = (N, \Sigma, P, S)$. $M_0 = (I_0 \cup \{q_s\}, V, q_s, I_f, \delta_0)$ *is the* ***nondeterministic LR(0) machine*** *for G where*

I_0*: set of 0-items,* V*: input alphabet,*

$q_s \notin I_0$*: initial state,* $I_f = I_0$*: set of final states,*

δ_0: $(I_0 \cup \{q_s\}) \times (V \cup \{\varepsilon\}) \to 2^{I_0}$ *of the form;*

$q_s\varepsilon \to [S \to \cdot\omega]$,

$[A \to \alpha\cdot X\beta]X \to [A \to \alpha X\cdot\beta]$, $X \in V$, *and*

$[A \to \alpha\cdot B\beta]\varepsilon \to [B \to \cdot\omega]$

where $\beta \Rightarrow^* w$, $^\exists w \in \Sigma^*$.

Nondeterministic LR(0) machine for G is computed in time $O(|G|^2)$ and of size $O(|G|^2)$.

Lemma 7.1 *The set of viable prefixes of G is the language accepted by the nondeterministic LR(0) machine M_0 for G, and for any viable prefix γ,*

$$Valid_0(\gamma) = \{q \mid {}^\exists q \in I_0 : q \in \delta_0'(q_s, \gamma)\}$$

where $\delta_0'(q, \alpha X) = \{p \mid {}^\exists \alpha \in V^*, X \in V, r \in \delta_0'(q, \alpha) : p \in \delta_0(r, X)\}$.

Proof. *By construction,*

$$Valid_0(\varepsilon) = {}^{1*}\{[S \to \cdot\omega] \mid S \to \omega \in P\} = \{q \mid {}^\exists q \in I_0 : q \in \delta_0'(q_s, \varepsilon)\}.$$

Assume as i.h. for γ of length n, n ≥ 0,

$I \neq \varnothing$.϶. $I = \{q \mid {}^\exists q \in I_0 : q \in \delta_0'(q_s, \gamma)\}$
iff γ *is viable prefix and* $I = Valid_0(\gamma)$.

$I' \neq \varnothing$.϶. $I' = \{q \mid {}^\exists q \in I_0 : q \in \delta_0'(q_s, \gamma X)\}$
iff γ *is viable prefix and* $I' = Goto(Valid_0(\gamma), X) \neq \varnothing$.
Since $Goto(Valid_0(\gamma), X) = Valid_0(\gamma X)$, *and*
$Valid_0(\gamma X) \neq \varnothing$ *iff* γX *is viable prefix,*
$I' \neq \varnothing$ *iff* γX *is viable prefix and* $I' = Valid_0(\gamma X)$.
Q.E.D.

Nondeterministic LR(0) machine has $O(|G|)$ states.

Theorem 7.2 *Deterministic LR(0) machine for any grammar G can be computed in time* $O(2^{|G|+2log|G|})$.

Theorem 7.3 *The LR(0) parser of any grammar G can be constructed in time* $O(2^{|G|+2log|G|})$.

Actions of an SLR(1) parser:

(sa) $[\delta] \mid a \to [\delta][\delta a] \mid$

(ra) $[\delta][\delta X_1]...[\delta X_1...X_m] \mid a \to [\delta][\delta A] \mid a$

where $[A \to X_1...X_m\cdot] \in Valid_0(\delta X_1...X_m)$ *and*

$a \in Follow'_1(A)$.

Theorem 7.4 *The SLR(1) parser of any grammar G can be constructed in time* $O(2^{|G|+2log|G|+log|T|})$.

A grammar G is not SLR(1) iff for some state q

(1) $\exists [A_1 \to \omega_1\cdot], [A_2 \to \omega_2\cdot] \in q$ *where*

$Follow'_1(A_1) \cap Follow'_1(A_2) \neq \emptyset$, *or*

(2) $\exists [A_1 \to \alpha\cdot a\beta], [A_2 \to \omega\cdot] \in q$ *where*

$a \in Follow'_1(A_2)$.

SLR(1) test can be performed in time

$O(|G|^2 \cdot 2^{|G|+log|G|}) = O(2^{|G|+3log|G|})$.

7.2 Construction of Canonical LR(1) Parsers

$[A \to \alpha\cdot\beta, \{a_1, \ldots, a_n\}]$ *represents;*

set of 1-items $\{[A \to \alpha\cdot\beta, a_1], \ldots, [A \to \alpha\cdot\beta, a_n]\}$.

Let $G = (N, \Sigma, P, S)$. $M_1 = (I_1 \cup \{q_s\}, V, q_s, I_f, \delta_1)$ *is the* ***nondeterministic LR(1) machine*** *for G where*

I_1: *set of 1-items,* V: *input alphabet,*

$q_s \notin I_1$: *initial state,* $I_f = I_1$: *set of final states,*

$\delta_1: (I_1 \cup \{q_s\}) \times (V \cup \{\varepsilon\}) \to 2^{I_1}$ *of the form;*

$q_s\varepsilon \to [S \to \cdot\omega, \varepsilon]$,

$[A \to \alpha\cdot X\beta, y]X \to [A \to \alpha X\cdot\beta, y], X \in V$,

$[A \to \alpha\cdot B\beta, y]\varepsilon \to [B \to \cdot\omega, z]$

where $\beta \Rightarrow^* w, \exists w \in \Sigma^*$, *and* $z \in First_1(\beta y)$.

Nondeterministic LR(1) machine for G is computed in time $O(|T|^2 \cdot |G|^2)$ *and of size* $O(|T|^2 \cdot |G|^2)$.

Lemma 7.5 *The set of viable prefixes of G is the language accepted by the nondeterministic LR(1) machine* M_1 *for G, and for any viable prefix* γ,

$$Valid_1(\gamma) = \{q \mid \exists q \in I_1: q \in \delta_1'(q_s, \gamma)\}$$

where $\delta_1'(q, \alpha X) = \{p \mid \exists \alpha \in V^*, X \in V, r \in \delta_1'(q, \alpha): p \in \delta_1(r, X)\}$.

Theorem 7.6 *The deterministic LR(1) machine for any grammar G can be computed in time*

$$O(2^{|G|^2+4\log|G|}).$$

Theorem 7.7 *The canonical LR(1) parser of any grammar G can be constructed in time*

$$O(2^{|G|^2+4\log|G|}).$$

A grammar G is not SLR(1) iff for some state q

(1) $\exists [A_1 \to \omega_1\cdot, a], [A_2 \to \omega_2\cdot, a] \in q$, *or*

(2) $\exists [A_1 \to \alpha\cdot a\beta, b], [A_2 \to \omega\cdot, a] \in q$ *where*
$a \in \Sigma \cup \{\$\}$.

Theorem 7.8 *A grammar G with terminal alphabet* Σ *can be tested for the LR(1) property in deterministic time* $O(2^{|G|^2+4\log|G|})$.

7.3 Construction of LALR(1) Parsers

To construct LALR(1) parser:

construct LR(0) collection (I_0),

add lookahead symbols in appropriate 0-items.

Let $G = (N, \Sigma, P, S)$ be a reduced grammar and $G' = (N \cup \{S'\}, \Sigma \cup \{\$\}, P \cup \{S' \rightarrow \$S\$\}, S')$ its \$-augmented grammar.

For state q in the LR(0) collection for G' and production rule $A \rightarrow \omega$ of G:

$$LALR(q, A \rightarrow \omega) = \{a \in \Sigma \cup \{\$\} \mid \exists \gamma: q = Valid_0(\gamma), [A \rightarrow \omega\cdot, a] \in Valid_1(\gamma)\}$$

where $LALR(q, A \rightarrow \omega)$ is ***LALR(1) lookahead set for the reduce action by rule $A \rightarrow \omega$ at state q****.*

LALR(1) lookahead sets can be determined directly from the transitions of the deterministic LR(0) machine.

Let Q be the set of states of the deterministic LR(0) machine for the \$-augmented grammar G' for grammar $G = (N, \Sigma, P, S)$.

$(q, A) \in Q \times N$ *where* $\exists p \in I_0, A \in N: q \xrightarrow{A} p$,

$(q, A \rightarrow \omega) \in Q \times P$ *where* $[A \rightarrow \omega\cdot] \in q$.

(q, A) **goes-to** *Goto(q, A), if* $\exists p \in I_0, A \in N: q \xrightarrow{A} p$;

q **has-transition-on** *X, if* $\exists p \in I_0, X \in V: q \xrightarrow{X} p$;

q **has-null-transition** *(q, A), if* $\exists p \in I_0, A \in N: q \xrightarrow{A} p \wedge A \Rightarrow^{+} \varepsilon$;

(Goto(q, α), A) **includes** *(q, B), if* $\exists p \in I_0, A,B \in N: q \xrightarrow{B} p \wedge B \to \alpha A \beta \in P, \beta \Rightarrow^{*} \varepsilon$;

(Goto(q, ω), A→ω) **lookback** *(q, A), if* $\exists p \in I_0, A \in N: q \xrightarrow{A} p$;

Lemma 7.9 *Let q be a state in Q that has a transition on a nonterminal B and let A be a nonterminal, and α a string in* V^* *such that* $B \Rightarrow_{rm}^{n} \alpha A$. *Then*

Goto(q, αA) ≠ ø and (Goto(q, α), A) **includes*** *(q, B).*

Proof. *By induction on n*

1) base: clear

2) $\exists m < n, \delta' \in V^*, A' \to \alpha' A \beta'$:

$B \Rightarrow_{rm}^{m} \delta' A' \Rightarrow_{rm} \delta' \alpha' A \beta' = \alpha A \beta', \beta' \Rightarrow^{*} \varepsilon$.

As i.h. Goto(q, δ'A') ≠ ø and

(Goto(q, δ'), A') **includes*** *(q, B).*

Then Goto(q, δ'α'A) ≠ ø and

(Goto(q, δ'α'), A) **includes** *(Goto(q, δ'), A').*

Q.E.D.

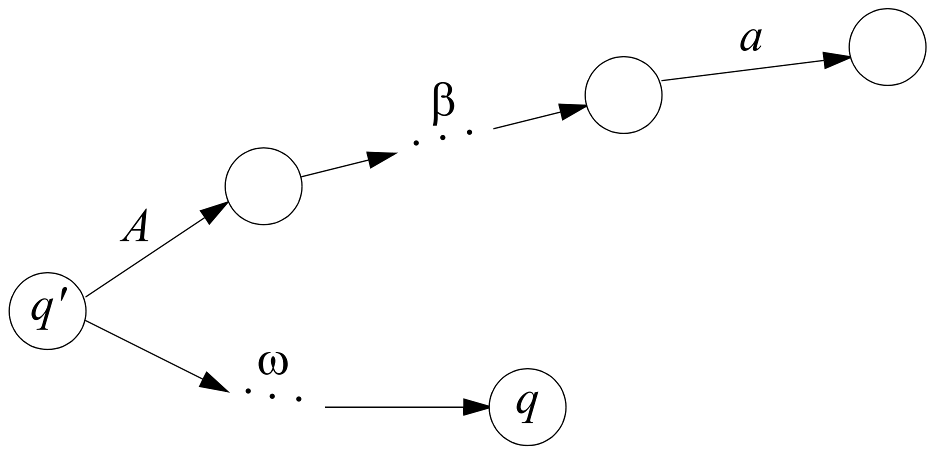
Lemma 7.10 *If for* $n \geq 0$, (q, A) ***includes***^{n} (q', B) *then for some* α *in* V^*, $q = Goto(q', \alpha)$ *and* $B \Rightarrow^{*}_{rm} \alpha A$.

***directly-reads* = *goes-to has-transition-on terminal*,**
***reads* = *goes-to has-null-transition*.**

Fact 7.11 *For* $n > 0$, (q, A) ***reads***^{n} (q', B)
iff B is nullable and there is a nullable string δ *of length* $(n\text{-}1)$ *such that* $q' = Goto(q, A\delta)$ *and* q' *has a transition on B.*

has-LALR-lookahead* = *lookback includes* ***reads****
***directly-reads*.**

$(q, A \rightarrow \omega)$ ***has-LALR-lookahead*** *a iff*
$a \in LALR(q, A \rightarrow \omega)$. *In other words,*
$LALR(q, A \rightarrow \omega)$ = ***has-LALR-lookahead*** $(q, A \rightarrow \omega)$.

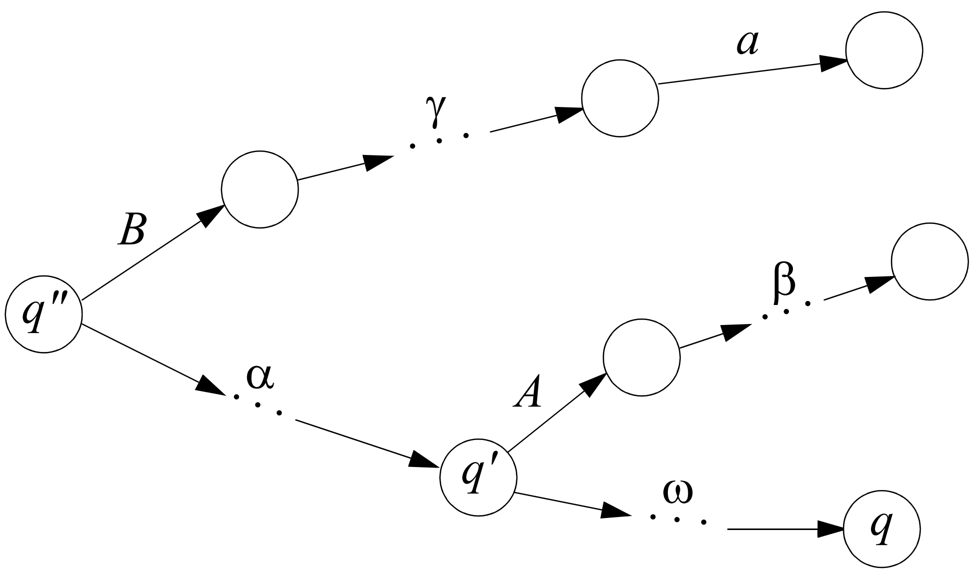


$A \rightarrow \omega \in P,\ \beta \Rightarrow^*_{rm} \varepsilon,$

$\exists \gamma, \omega \in V^*: Valid_0(\gamma\omega) = q,\ [A \rightarrow \omega\cdot, a] \in Valid_1(\gamma\omega),$

$a \in LALR(q, A \rightarrow \omega),$

$(q, A \rightarrow \omega)$ ***lookback*** (q', A) ***reads**** ***directly-reads*** a.



$A \rightarrow \omega \in P,\ B \rightarrow \alpha A \beta \in P,\ \beta \Rightarrow^*_{rm} \varepsilon,\ \gamma \Rightarrow^*_{rm} \varepsilon,$

$a \in LALR(q, A \rightarrow \omega),$

$(q, A \rightarrow \omega)$ ***lookback*** (q', A) ***includes*** (q'', B) ***reads**** ***directly-reads*** a.

Lemma 7.12 *If* $\gamma\beta$ *is a viable prefix of a reduced grammar and* $\beta \Rightarrow_{rm}^{n} az$*, where* $n \geq 0$*, a is a terminal and z is a terminal string, then there is a viable prefix* $\gamma\delta a$*, where* δ *is nullable.*

Proof*. By induction on n.*

1) base: $\beta = \delta a\psi$*, where* $\delta \Rightarrow_{rm}^{*} \varepsilon$.

2) $\beta = \delta' B\psi$*, where* $\delta' \Rightarrow_{rm}^{*} \varepsilon$*,* $B \Rightarrow \beta' \Rightarrow_{rm}^{n-1} az'$.

Since $\gamma\delta' B$ *is viable,* $\gamma\delta'\beta'$ *is also viable.*

As $\beta' \Rightarrow_{rm}^{n-1} az'$*, by applying i.h. to viable prefix* $\gamma'\beta'$*, where* $\gamma' = \gamma\delta'$*. Q.E.D.*

Lemma 7.13 *In a reduced grammar, (q, A)* ***reads**** ***directly-reads*** *a iff q contains an item* $[B \rightarrow \alpha \cdot A\beta]$ *with* $a \in First_1(\beta)$.

Proof. *Let* γ *be a viable prefix .϶.* $q = Valid_0(\gamma)$.

$\Rightarrow$: *Assume (q, A)* ***reads**** ***directly-reads*** *a,*

then $^\exists\delta \in V^*: \delta \Rightarrow^*_{rm} \varepsilon$, $Goto(q, A\delta a) \neq \emptyset$.

$\gamma A\delta a$ *is viable prefix of G' and*

$S' \Rightarrow^*_{rm} \gamma' By \Rightarrow_{rm} \gamma'\alpha'\beta' y = \gamma A\delta a\beta' y$, $B \rightarrow \alpha'\beta'$.

(1) *In case* $\alpha' = \alpha A\delta a$, *where* α *is suffix of* γ:

$S' \Rightarrow^*_{rm} \gamma' By \Rightarrow_{rm} \gamma'\alpha A\beta y = \gamma A\beta y$, *where* $\beta = \delta a\beta'$.

$\therefore [B \rightarrow \alpha \cdot A\beta] \in Valid_0(\gamma)$ *and* $a \in First_1(\beta)$.

(2) *In case* $\gamma' = \gamma A\eta$ *and* $S' \Rightarrow^*_{rm} \gamma' By = \gamma A\eta By$, *where* $a \in First_1(\eta\beta)$. *Then*

$S' \Rightarrow^*_{rm} \delta' B' y' \Rightarrow_{rm} \delta'\alpha'' A\beta'' y' = \gamma A\beta'' y'$,

$\beta'' y' \Rightarrow^* \eta By$, $B' \rightarrow \alpha'' A\beta''$.

$\therefore [B' \rightarrow \alpha'' \cdot A\beta''] \in Valid_0(\gamma)$,

$\beta'' \Rightarrow^* \eta Bz$, *where z is prefix of y.* $a \in First_1(\beta'')$.

$\Leftarrow$: *Assume* $[B \rightarrow \alpha \cdot A\beta] \in q$ *and* $a \in First_1(\beta)$.

$\gamma A\beta$ *is a viable prefix,*

$\gamma A\delta a$ *is also viable prefix, where* $\delta \Rightarrow^* \varepsilon$.

$Goto(q, A\delta a) \neq \emptyset$.

$\therefore$ *(q, A)* ***reads**** ***directly-reads*** *a.*

Lemma 7.14 *Let G = (N, Σ, P, S) be a grammar. Further let A be a nonterminal, X and Y in V, γ and ψ strings in* V^**, y a string in* Σ^**, and π a rule string in* P^* *such that*

$$A \underset{rm}{\overset{\pi}{\Rightarrow}} \gamma X \psi Y y \text{ and } \psi \Rightarrow^* \varepsilon.$$

Then there are symbols X' and Y' in V, a rule r' = B → $\alpha X'\psi'Y'\beta$ *in P, and strings γ', α', β' in* V^* *and y' in* Σ^* *such that*

$$A \underset{rm}{\overset{\pi'}{\Rightarrow}} \gamma' B y' \underset{rm}{\overset{r'}{\Rightarrow}} \gamma'\alpha X'\psi'Y'\beta y',\ \gamma'\alpha\alpha' = \gamma,$$

$$X' \Rightarrow_{rm}^* \alpha' X,\ \psi' \Rightarrow_{rm}^* \varepsilon, \text{ and } Y' \Rightarrow_{lm}^* Y\beta',$$

where π'r' is a prefix of π. In other words, in the rightmost derivation of γXψYy from A there is a step showing that the symbols X and Y "originate" from a pair of adjoining symbols in the right-hand side of the same rule.

Lemma 7.15 *In a reduced grammar, terminal a belongs to LALR(q, A→ω) iff there is a rule* $C \to \alpha B\beta$ *and state q' such that*

$(q, A\to\omega)$ ***lookback includes***** (q', B),

$[C \to \alpha\cdot B\beta] \in q'$, and $a \in First_1(\beta)$.

Proof.

⇒: Assume $a \in LALR(q, A\to\omega)$, then $\exists\gamma$ of G', y:

$S' \Rightarrow^*_{rm} \gamma A a y \Rightarrow_{rm} \gamma\omega a y$, where $Valid_0(\gamma\omega) = q$.

$S' \Rightarrow^*_{rm} \gamma' C y' \Rightarrow_{rm} \gamma'\alpha B\beta y'$, $\gamma'\alpha\alpha' = \gamma$,

$B \Rightarrow^*_{rm} \alpha' A$, $\beta \Rightarrow^* a\beta'$,

$[C \to \alpha\cdot B\beta] \in Valid_0(\gamma'\alpha)=q'$, $a \in First_1(\beta)$,

$\gamma'\alpha B$ is a viable prefix.

∴ $Goto(q', \alpha' A) \neq \emptyset$, $(Goto(q', \alpha'), A)$ ***includes***** (q', B)

$Goto(q', \alpha') = Valid_0(\gamma'\alpha\alpha') = Valid_0(\gamma)$.

γA is a viable prefix, thus

$(Valid_0(\gamma, \omega), A\to\omega)$ ***lookback*** $(Valid_0(\gamma), A)$,

∴ $(q, A\to\omega)$ ***lookback includes***** (q', B).

$\Leftarrow$: *Assume* $(q, A \to \omega)$ ***lookback includes**** (q', B), $[C \to \alpha \cdot B\beta] \in q'$, $a \in First_1(\beta)$.

$S' \Rightarrow^*_{rm} \delta C y \Rightarrow_{rm} \delta\alpha B\beta y$, $Valid_0(\delta\alpha) = q'$.

Since $a \in First_1(\beta)$, $\beta \Rightarrow^*_{rm} az$.

$\therefore S' \Rightarrow^*_{rm} \delta\alpha B\beta y \Rightarrow^*_{rm} \delta\alpha Bazy$.

$((q, A \to \omega)$ ***lookback*** (q_1, A) ***includes**** (q', B), *where* $Goto(q_1, \omega) = q$.

$\therefore \exists \alpha'$, $q_1 = Goto(q', \alpha')$, $B \Rightarrow^*_{rm} \alpha' A$:

$S' \Rightarrow^*_{rm} \delta\alpha Bazy \Rightarrow^*_{rm} \delta\alpha\alpha' Aazy \Rightarrow_{rm} \delta\alpha\alpha'\omega azy$,

where $Valid_0(\delta\alpha\alpha'\omega) = Goto(q', \alpha'\omega) = Goto(q_1, \omega) = q$.

$\therefore a \in LALR(q, A \to \omega)$.

Theorem 7.16 *Let G be a reduced grammar and G' its \$-augmented grammar. Terminal a of G' is in the LALR(1) lookahead set for the reduce action by rule $A \rightarrow \omega$ of G at state q in the deterministic LR(0) machine for G' iff*

$(q, A\rightarrow\omega)$ ***has-LALR-lookahead*** *a.*

Theorem 7.17 *Let DM be the deterministic LR(0) machine for the \$-augmented grammar G' for a reduced grammar G. The collection of all LALR(1) lookahead sets LALR$(q, A\rightarrow\omega)$, where q is a state of DM and $A \rightarrow \omega$ is a rule of G, can be computed in time $O(t\cdot|G|\cdot|Q|)$, where Q is the set of states of DM and t is the time taken by one set operation (assignment or union) on subsets of Σ.*

Fact 7.18 *Reduce actions of an LALR(1) parser:*

$$[\delta]_0[\delta X_1]_0 \ldots [\delta X_1 \ldots X_m]_0 \mid a \rightarrow [\delta]_0[\delta A]_0 \mid a,$$

where $A \rightarrow X_1 \ldots X_m \in P$, δA is a viable prefix of G', and $a \in LALR(Valid_0(\delta X_1 \ldots X_m), A \rightarrow X_1 \ldots X_m)$.

Theorem 7.19 *The LALR(1) parser of any grammar G can be constructed in time*

$$O(2^{|G|+2log|G|+log|T|}).$$

LALR(1) test can be performed in time

$$O(2^{|G|+2log|G|+log|T|}).$$

7.5 Optimization of LR(1) Parsers

Inessential Error Entries

$^{\exists}$q, a: Action[q,a] = "error",

*(q, a) is an **essential error entry**, if*

$^{\exists}w,y \in \Sigma^{*}, \phi \in Q^{*}$: $\$[\$] \mid w\$ \underset{M}{\Rightarrow}^{*} \$\phi q \mid y\$$, $1{:}y\$ = a$.

*Otherwise (q, a) is **inessential**.*

Fact 7.24 *For state q = [γb], where γb is a viable prefix ending with a terminal, all error entries (q,a), $a \in \Sigma \cup \{\$\}$, are essential.*

Fact 7.25 *Let q = [δA] for some viable prefix ending with a nonterminal in $\Sigma \cup \{\$\}$ such that (q,a) is an error entry. The error entry (q, a) is essential iff*

$$\$\phi' q' \mid ay \underset{M}{\Rightarrow}^{*} \$\phi q \mid ay,$$

where q' = [γb] for some terminal b in $\Sigma \cup \{\$\}$, and the configuration $\$\phi' q' \mid ay$ is accesible from some initial configuration.

$(q, A \rightarrow \alpha{\cdot}B\beta)$, $(q, {\cdot}B)$, $(q, B{\cdot})$

$(Goto[q,\alpha], B{\cdot})$ **symbol-in** $(q, A \rightarrow \alpha B{\cdot}\beta)$
$(q, A \rightarrow \alpha{\cdot}B\beta)$ **points** $(Goto[q,\alpha], {\cdot}B)$
$(q, {\cdot}B)$ **expands** $(q, B \rightarrow {\cdot}\omega)$
$(q, A \rightarrow \alpha X{\cdot}\beta)$ **entered-by** X, where $X \in V$

$(q, B \rightarrow \omega{\cdot})$ **on-*a*-reduce-to** $(q, B{\cdot})$ whenever
$Action[Goto[q, \omega], a]$ = "reduce by $B \rightarrow \omega$"

$(q, A \rightarrow \alpha{\cdot}B\beta)$ **directly-on-*a*-passes-null**
$(q, A \rightarrow \alpha B{\cdot}\beta)$ whenever
$Action[Goto[q, \alpha], a]$ = "reduce by $B \rightarrow \varepsilon$"

$(q, A \rightarrow \alpha{\cdot}\beta)$ **error-entry-on**-*a* $(Goto[q, \alpha], a)$
whenever $Action[Goto[q, \alpha], a]$ = "error"

directly-descends* = *points expands

***may-on-*a*-access* = (*on-*a*-reduces-to symbol-in* $\cup$ *directly-descends** ·*directly-on-*a*-passes-null*)***

***may-imply-*a*-essential* = *terminal entered-by*$^{-1}$ ·*may-on-*a*-access error-entry-on-*a*.**

Lemma 7.28 *Let* $n \geq 0$ *and*

$$\$[X_1][X_1X_2] \ldots [X_1 \ldots X_m] \mid a \overset{n}{\underset{M}{\Rightarrow}}$$
$$\$[Y_1][Y_1Y_2] \ldots [Y_1 \ldots Y_p] \mid a .$$

Then for all $[B \rightarrow Y_{j+1} \ldots Y_p \cdot \beta] \in Valid(Y_1 \ldots Y_p), j < p$, *there is an item* $[A \rightarrow X_{i+1} \ldots X_m \cdot \alpha] \in Valid(X_1 \ldots X_m)$ *such that*

$([X_1 \ldots X_i], A \rightarrow X_{i+1} \ldots X_m \cdot \alpha)$ ***may-on-a-access***
$([Y_1 \ldots Y_j] , B \rightarrow Y_{j+1} \ldots Y_p \cdot \beta)$.

Proof. *By induction on n.*

1) base: trivial

2) Induction:

i) case 1.

$$\$[X_1] \ldots [X_1 \ldots X_m] \mid a \overset{n-1}{\underset{M}{\Rightarrow}} \$[Y_1] \ldots [Y_1 \ldots Y_{p-1}] \mid a$$
$$\Rightarrow \$[Y_1] \ldots [Y_1 \ldots Y_p] \mid a$$

i-1) hypothesis:

$\forall [C \rightarrow Y_{k+1} \ldots Y_{p-1} \cdot \gamma] \in Valid(Y_1 \ldots Y_{p-1}), k<(p-1)$
$\exists [A \rightarrow X_{i+1} \ldots X_m \cdot \alpha] \in Valid(X_1 \ldots X_m)$,
$([X_1 \ldots X_i], A \rightarrow X_{i+1} \ldots X_m \cdot \alpha)$ ***may-on-a-access***
$([Y_1 \ldots Y_k] ,C \rightarrow Y_{k+1} \ldots Y_{p-1} \cdot \gamma)$,

i-2) step:

$Action[[Y_1 \ldots Y_{p-1}],a]=Y_p \rightarrow \cdot$, $[Y_p \rightarrow \cdot] \in Valid(Y_1 \ldots Y_{p-1})$

$([Y_1 \ldots Y_k] ,C \rightarrow Y_{k+1} \ldots Y_{p-1} \cdot \gamma)$

directly-descends* ***directly-on-a-passes-null***

$([Y_1 \ldots Y_j]\ ,\ B \rightarrow Y_{j+1} \ldots Y_p \cdot \beta)$

ii) case 2.

$\$[X_1] \ldots [X_1 \ldots X_m] \mid a \overset{n}{\underset{M}{\Rightarrow}} \$[Y_1] \ldots [Y_1 \ldots Y_{p-1} Z_1 \ldots Z_l] \mid a$

$\Rightarrow \$[Y_1] \ldots [Y_1 \ldots Y_p] \mid a$

ii-1) hypothesis:

$\forall [\beta], B \rightarrow \gamma \cdot \delta$ *such that* $\beta\gamma = Y_1 \ldots Y_{p-1} Z_1 \ldots Z_l$, $|\gamma| \geq 1$, *and* $[B \rightarrow \gamma \cdot \delta] \in Valid(\beta\gamma)$,

$\exists\ [A \rightarrow X_{i+1} \ldots X_m \cdot \alpha] \in Valid(X_1 \ldots X_m)$,

$([X_1 \ldots X_i], A \rightarrow X_{i+1} \ldots X_m \cdot \alpha)$ ***may-on-a-access***

$([\beta], B \rightarrow \gamma \cdot \delta)$

ii-2) step:

$Action[[Y_1 \ldots Y_{p-1} Z_1 \ldots Z_l], a] = Y_p \rightarrow Z_1 \ldots Z_l$

$[Y_p \rightarrow Z_1 \ldots Z_l \cdot] \in Valid(Y_1 \ldots Y_{p-1} Z_1 \ldots Z_l)$,

$([X_1 \ldots X_i]\ , A \rightarrow X_{i+1} \ldots X_m \cdot \alpha)$ ***may-on-a-access***

$([Y_1 \ldots Y_{p-1}]\ ,\ Y_p \rightarrow Z_1 \ldots Z_l \cdot)$

$([Y_1 \ldots Y_{p-1}], Y_p \rightarrow Z_1 \ldots Z_l \cdot)$ ***on-a-reduce-to symbol-in***

$([Y_1 \ldots Y_j]\ ,\ B \rightarrow Y_{j+1} \ldots Y_p \cdot \beta)$

Lemma 7.29 *If an error (q,a) is essential, then b* ***may-imply-a-essential*** *(q,a) for some* $b \in \Sigma \cup \{\$\}$.

Lemma 7.30 *The set of essential error entries is included in the set* ***may-imply-essential*** $(\Sigma \cup \{\$\})$.

$(q, B\cdot)$ ***left-corner-in*** $(q, A \to B\cdot\beta)$

$(q, A \to \alpha\cdot B\beta)$ ***on-a-passes-null*** $(q, A \to \alpha B\cdot\beta)$

if $(q, A \to \alpha\cdot B\beta)$ ***may-on-a-access*** $(q, A \to \alpha B\cdot\beta)$

on-a-access = (on-a-reduces-to symbol-in ∪ on-a-passes-null)* $\cdot$ ***(on-a-reduces-to left-corner-in ∪ directly-descends ∪ on-a-passes-null)****

imply-a-essential = terminal entered-by^{-1} $\cdot$ ***on-a-access error-entry-on-a***

Lemma 7.34 *Let* $n \geq 0$ *and*

$\$[X_1] \dots [X_1 \dots X_m] \mid a \underset{M}{\overset{n}{\Rightarrow}} \$[Y_1] \dots [Y_1 \dots Y_p] \mid a.$

Then for all $[B \to Y_{j+1} \dots Y_p\cdot\beta] \in Valid(Y_1 \dots Y_p)$, $j<p$ *there is an item* $[A \to X_{i+1} \dots X_m\cdot\alpha] \in Valid(X_1 \dots X_m)$ *such that*

$([X_1 \dots X_i], A \to X_{i+1} \dots X_m\cdot\alpha)$ ***on-a-access*** $([Y_1 \dots Y_j], B \to Y_{j+1} \dots Y_p\cdot\beta)$.

Lemma 7.35 *Let* $Y_1 \dots Y_p$, $p>0$, *be a viable prefix of* G' *and* $[B \to Y_{j+1} \dots Y_p\cdot\beta]$, $j<p$, *be an item in* $Valid(Y_1 \dots Y_p)$. *If*

$([Y_1 \dots Y_j], B \to Y_{j+1} \dots Y_p\cdot Z\beta)$ ***on-a-passes-null*** $([Y_1 \dots Y_j], B \to Y_{j+1} \dots Y_p Z\cdot\beta)$, *then*

$$\$[Y_1] \dots [Y_1 \dots Y_p] \mid a \underset{M}{\overset{*}{\Rightarrow}} \$[Y_1] \dots [Y_1 \dots Y_p][Y_1 \dots Y_p Z] \mid a.$$

Lemma 7.36 *Let $n \geq 0$, and*

*$(q, A \rightarrow \alpha\cdot\beta)$ (**on-a-reduces-to symbol-in** $\cup$ **on-a-passes-null**)^n $(q', B \rightarrow \gamma\cdot\delta)$*

where $\alpha \neq \varepsilon$ and $\gamma \neq \varepsilon$. Then for any viable prefix $Y_1 \ldots Y_p$, $p \geq 1$, of G' and $j < p$ such that

$[Y_1 \ldots Y_j] = q'$ *and* $Y_{j+1} \ldots Y_p = \gamma$,

there is a viable prefix $X_1 \ldots X_m$, $m \geq 1$, and $i < m$ such that

$X_1 \ldots X_i = q$, $X_{i+1} \ldots X_m = \alpha$ *and*

$\$[X_1] \ldots [X_1 \ldots X_m] \mid a \overset{*}{\underset{M}{\Rightarrow}} \$[Y_1] \ldots [Y_1 \ldots Y_p] \mid a.$

Proof. *By induction on n.*

1) base: trivial (n = 0)

2) hypothesis:

i) case 1.

*$(q, A \rightarrow \alpha\cdot\beta)$ (**on-a-reduces-to symbol-in** $\cup$ **on-a-passes-null**)$^{n-1}$ $(q, C \rightarrow \omega\cdot)$ **on-a-reduces-to symbol-in** $(q', B \rightarrow \gamma\cdot\delta)$*

$\exists$viable prefix $X_1 \ldots X_m$, $m \geq 1$, of G' and $i < m$ such that

$X_1 \ldots X_i = q$, $X_{i+1} \ldots X_m = \alpha$ *and*

$\$[X_1] \ldots [X_1 \ldots X_m] \mid a \overset{*}{\underset{M}{\Rightarrow}} \$[Z_1] \ldots [Z_1 \ldots Z_l] \mid a$

for a viable prefix $Z_1 \ldots Z_l$ and $k < l$ such that

$[Z_1 \ldots Z_k] = q''$, $Z_{k+1} \ldots Z_l = \omega$.

i-1) step:

By the def. of ***on-a-reduces-to symbol-in***

for any viable prefix $Y_1 \ldots Y_p$, $p \geq 1$, *of G' and* $j<p$ ∍

$[Y_1 \ldots Y_p] = q'$ *and* $Y_{j+1} \ldots Y_p = \gamma$

∃ *a viable prefix* $Z_1 \ldots Z_l$ *and* $k<l$ *such that*

$[Z_1 \ldots Z_k] = q''$, $Z_{k+1} \ldots Z_l = \omega$ *and*

$\$[Z_1] \ldots [Z_1 \ldots Z_m] \mid a \underset{M}{\Rightarrow}^{*} \$[Y_1] \ldots [Y_1 \ldots Y_p] \mid a.$

$\therefore$ $X_1 \ldots X_i = q$, $X_{i+1} \ldots X_m = \alpha$ *and*

$\$[X_1] \ldots [X_1 \ldots X_m] \mid a \underset{M}{\Rightarrow}^{*} \$[Y_1] \ldots [Y_1 \ldots Y_p] \mid a.$

ii) case 2.

$(q, A \rightarrow \alpha \cdot \beta)$ ***(on-a-reduces-to symbol-in*** ∪ ***on-a-passes-null)***$^{n-1}$ $(q', B \rightarrow \gamma' \cdot Y\delta)$ ***on-a-passes-null*** $(q', B \rightarrow \gamma' Y \cdot \delta) = (q', B \rightarrow \gamma \cdot \delta)$

By hypothesis and ***Lemma 7.35***.

Lemma *7.37 Let n Š 0, and*
*(q, A → α·β) **(on-a-reduces-to left-corner-in** ∪ **directly-descends** ∪ **on-a-passes-null)**^{n} (q',B → γ·δ)*
Then there is a vaible prefix $Y_1 \dots Y_p$, p≥1, of G' and j≤p,k ≤j such that

$[Y_1 \dots Y_j] = q', Y_{j+1} \dots Y_p = \gamma, [Y_1 \dots Y_k] = q$ and

$\$[Y_1] \dots [Y_1 \dots Y_k\alpha] \mid a \underset{M}{\Rightarrow}^* \$[Y_1 \dots Y_p] \mid a$.

Proof. *By induction on n.*

1) base: (n = 0) q = q' , A → α·β = B → γ·δ

2) induction:

i) case 1: (j+1) = p and

*(q, A → α·β) **(on-a-reduces-to left-corner-in** ∪ **directly-descends** ∪ **on-a-passes-null)**$^{n-1}$*
*(q', C → ω·) **on-a-reduces-to left-cornet-in***
(q', B → C·δ) = (q', B → γ·δ)

i-1) hypothesis: ∃viable prefix $Y_1 \dots Y_r$ and j ≤ r, k ≤ j э

$[Y_1 \dots Y_j] = q', Y_{j+1} \dots Y_r = \gamma, [Y_1 \dots Y_k] = q$ and

$\$[Y_1] \dots [Y_1 \dots Y_k\alpha] \mid a \underset{M}{\Rightarrow}^* \$[Y_1 \dots Y_r] \mid a$.

$\therefore \$[Y_1] \dots [Y_1 \dots Y_r] \mid a$

$= [Y_1] \dots [Y_1 \dots Y_k] \dots [Y_1 \dots Y_j] \dots [Y_1 \dots Y_j\omega] \mid a$

$\underset{M}{\Rightarrow} [Y_1] \dots [Y_1 \dots Y_k] \dots [Y_1 \dots Y_j] \dots [Y_1 \dots Y_j\omega] \mid a$

ii) case 2: $\gamma = \varepsilon$ *and*

$(q, A \rightarrow \alpha\cdot\beta)$ ***(on-a-reduces-to left-cornet-in*** $\cup$ ***directly-descends*** $\cup$ ***on-a-passes-null)***$^{n-1}$

$(q'', C \rightarrow \eta\cdot\psi)$ ***directly-descends***

$(Goto[q'',\eta], B \rightarrow \cdot\delta) = (q', B \rightarrow \gamma\cdot\delta)$

where $[C \rightarrow \eta\cdot\psi] \in Goto[q'',\eta]$

ii-1) hypothesis: $^{\exists}$*viable prefix* $Y_1 \ldots Y_p$ *and* $l \leq p$, $k \leq l \ni$

$[Y_1 \ldots Y_l] = q''$, $Y_{l+1} \ldots Y_p = \eta$, $[Y_1 \ldots Y_k] = q$ *and*

$\$[Y_1] \ldots [Y_1 \ldots Y_k\alpha] \mid a \underset{M}{\Rightarrow}^{*} \$[Y_1 \ldots Y_p] \mid a$.

$\therefore$ *when* $j = p$, *since*

$q' = Goto[q'',\eta] = [Y_1 \ldots Y_p]$ *and* $\gamma = \varepsilon$

iii) case3: Exercise

$(q, A \rightarrow \alpha\cdot\beta)$ ***(on-a-reduces-to left-cornet-in*** $\cup$ ***directly-descends*** $\cup$ ***on-a-passes-null)***$^{n-1}$

on-a-passes-null $(q', B \rightarrow \gamma\cdot\delta)$

Lemma 7.38 *Let q and q' be state of M and* $[A \rightarrow \alpha \cdot \beta]$ *and* $[B \rightarrow \gamma \cdot \delta]$ *items such that*

$$(q, A \rightarrow \alpha \cdot \beta) \textbf{ on-}a\textbf{-passes-null } (q', B \rightarrow \gamma \cdot \delta).$$

Then there are viable prefixes $X_1 \ldots X_m$, *mŠ1,* $Y_1 \ldots Y_p$, $p \geq 1$, *of G' and* $i \leq m$ *and* $j \leq p$ *such that*

$$[X_1 \ldots X_i] = q,\ X_{i+1} \ldots X_m = \alpha$$

$$[Y_1 \ldots Y_j] = q',\ Y_{j+1} \ldots Y_p = \gamma \text{ and}$$

$$\$[X_1] \ldots [X_1 \ldots X_m] \mid a \underset{M}{\overset{*}{\Rightarrow}} \$[Y_1] \ldots [Y_1 \ldots Y_p] \mid a.$$

Proof: *By* ***Lemma*** *7.36 and 7.37.*

Lemma 7.39 *An error entry (q,a) is* ***essential*** *iff b* ***implies-****a****-essential*** *(q,a) for some* $b \in \Sigma \cup \{\$\}$.
Proof: *By* ***Lemma*** *7.34, 7.38 and* ***Fact 7.25***.

implies-essential = ***implies-***a_1***-essential*** $\cup \ldots \cup$ ***implies-***a_n***-essential***.

where $\{a_1, \ldots, a_n\} = \Sigma \cup \{\$\}$

Lemma 7.40 *The set of essential error entries is obtained as the set* ***implies-essential*** $(\Sigma \cup \{\$\})$.
Proof: *By* ***Lemma*** *7.39.*

Reducing the number of state in an LR(1) parser

*For a deterministic LR(0)-based LR(1) parser M of G, two state q_1 and q_2 are **compatible**, if*

(1) $Action[q_1, a] = Action[q_2, a]$, or either (q_1, a) or (q_2, a) is an inessential error entry, for all $a \in \Sigma \cup \{\$\}$

(2) $Goto[q_1, A] = Goto[q_2, A]$, or either (q_1, A) or (q_2, A) is an error entry, for all $A \in N$

Let [G'] be the set of states of M

*$\rho = \{Q_1, ..., Q_m\}$ is a **compatible partition** of [G'] if Q_i in ρ contains only pairwise compatible states*

***Theorem 7.43** Let $G = (N, \Sigma, P, S)$ be an LALR(1) grammar and M its LALR(1) parser. Let ρ be a compatible partition of the set of state of M, and let Action' and Goto' be tables defined by:*

$\forall Q \in r, a \in S \cup \{\$\}$:

$Action'[Q,a]$ = error, if $\forall (q,a), q \in Q$, are error
= $Action[q,a]$, where $q \in Q$ and (q,a) is not error entry, otherwise;

$\forall X \in V, Goto'[Q,X] = \bigcup_{q \in Q} Goto[q,X]$

Then Action' and Goto' form a parsing table that represents a right parser of G which behave in the same way as M.

Eliminating reduction by unit rules

Let G be an LALR(1) grammar and M be its parser. Let $A \rightarrow B$ be a unit rule of G.

q_1= Goto[q, A] and q_2= Goto[q, B] are (A,B)-compatible, if

(1) for all $a \in \Sigma \cup \{\$\}$, Action[$q_1$, a] = Action[$q_2$, a] or one of the three is true

(a) Action[q_2, a] = reduce by $A \rightarrow B$

(b) (q_2, a) is an inessential error entry, or

(c) (q_1, a) is an inessential error entry

(2) for all $C \in N$, one of the following statement is true

(d) Goto[q_1, C] = Goto[q_2, C],

(e) Goto[q_1, C] = ø, or

(f) Goto[q_2, C] = ø.

If q_1, q_2 are (A,B)-compatible,

(1) replace Goto[q, B] by q_1.

$\forall a \in \Sigma \cup \{\$\}$, whenever Action[$q_1$, a] = "error" and Action[q_2, a] $\neq$ "reduce by $A \rightarrow B$",

(2) replace Action[q_1, a] by Action[q_2, a].

Theorem 7.45 *Resulting parser behaves exactly in the same way as the original parser, except that it possibley bypass some reductions by unit rules.*

Let G be an LALR(1) grammar and M be its parser. Let $A_1 \rightarrow A_2$, $A_2 \rightarrow A_3$, ... $A_{p-1} \rightarrow A_p$ be a sequence of unit rules of G.
q_1= Goto[q, A_1] , ... , q_p= Goto[q, A_p]
Assume q_1 no reduction by a unit rule

Lemma 7.46 *Let A_1, A_2, ..., A_p and q_1, ... , q_p as above. For all i, i=1, ... , p-1, and for all a $\in \Sigma \cup \{\$\}$, if ($q_i$) is not an error entry, then Action[q_{i+1},a] = "reduce by $A_i \rightarrow A_{i+1}$"*

Lemma 7.47 *Let A_1, A_2, ..., A_p and q_1, ... , q_p as above. For any two distinct state q_i and q_j and for any $X \in V$, Goto[q_i, X] = ø or Goto[q_j, X] = ø.*

Each new state q_i', $2 \le i \le p$ is defined by extending Action and Goto table as follows.

(1) for all $a \in \Sigma \cup \{\$\}$,
$Action[q_i', a]$ =

1) $Action[q_1, a]$ if $Action[q_1, a]$ is not error
2) $Action[q_j, a]$ where $1 < j < i$, if
   $Action[q_j, a]$ not in {"error",
   "reduced by $A_{j-1} \to A_j$"}
3) $Action[q_1, a]$ otherwise

(2) for all $X \in N$,
$Goto[q_i', X]$ =

$Goto[q_j, X]$, where $1 \le j < i$,
   if $Goto[q_j, X]$ is not ø
$Goto[q_i, X]$, otherwise