Topics covered:

- Viable Prefixes
- Valid Items

Viable Prefixes:

Consider the CFG:

E -> E + T | T

T -> T * F | F

F -> id

Let w = id * id (input string)

Here is the trace of the SLR parsing algorithm:

STACK	INPUT	ACTION
$	id * id $	shift
$ id	* id $	reduce
$ F	* id $	reduce
$ T	* id $	shift
$T *	id $	shift
$T * id	$	reduce
$T * F	$	reduce
$ T	$	reduce
$ E	$	ACCEPT

We observe that at any point of time, the stack contents must be a prefix of a right sentential form. However, not all prefixes of a right sentential form can appear on the stack.

For example, consider the rightmost derivation:

Rightmost derivation of id * id	Set of prefixes of a right sentential form	Viable Prefixes
E -> T	ϵ, T	T
-> T * F	ϵ, T, T * , T * F	T, T *, T * F
-> T * id	ϵ, T, T * , T * id	T, T * , T * id
-> F * id	ϵ, F, F * , F * id	F
-> **id *** id	ϵ, id, id *, id * id	id

Here, ‘id *’ is a prefix of a right sentential form. But it can never appear on the stack! This is because we will always reduce by F -> id before shifting ‘*’

<u>Definition (viable prefix):</u> The prefixes of right sentential forms that can appear on the stack of a shift-reduce parser are called viable prefixes. Its a building block for recognizing handles.

By definition, a viable prefix is a prefix of a right sentential form that does not continue past the right end of the rightmost handle of that sentential form. It’s a viable prefix because it is a prefix of the handle.

Example:

Let: S -> X1 X2 X3 X4
A -> X1 X2

Let w = X1 X2 X3

SLR parse trace:

STACK	INPUT
$	X1 X2 X3 $
$ X1	X2 X3 $
$ X1 X2	X3 $
$ A	X3 $
$ A X3	$

.

.

As we see, X1 X2 X3 will never appear on the stack. So, it is not a viable prefix.

Importance of Viable Prefixes:

The entire SLR parsing algorithm is based on the idea that the LR(0) automaton can recognize viable prefixes and reduce them appropriately.

Recognizing Viable Prefixes Idea: To recognize viable prefixes, we must
–Recognize a sequence of partial rhs's of productions, where
–Each sequence can eventually reduce to part of the missing suffix of its predecessor.

Valid Items:

Consider the item:
A -> β 1 . β 2

Let S -> α A β
    -> α β 1 β 2 B (rightmost derivation, sentential form) .

Since the dot is between β 1 and β 2, α β 1 will be on top of the stack.

So, α β 1 is a viable prefix.

We say that A -> β 1 . β 2 is a ***valid item*** for the viable prefix α β 1.

Every viable prefix is associated with a valid item. In general, an item will be valid for many viable prefixes.