Definitions

Given a grammar $G = \{N, \Sigma, P, S\}$ where N are the non-terminals, Σ the terminals, P the production rules and S the “starting” or **goal symbol** of the grammar:

Item
: A production rule of G with a “progress marker” $\bullet$

Fresh Start of A
: An item of G with A on the LHS and the progress marker before all RHS symbols
  eg: $A \to \bullet\alpha\beta\pi$

Item Set
: A (sub)collection of **items** for a grammar G

(LR Parser) State
: Synonymous with **Item Set**

Closure of an Item Set — Prose

1. Make a copy of the item set, call it C
2. For every item with a non-terminal B to the right of $\bullet$, add all the **fresh starts** of B to C
3. Repeat step 2 until C is unchanged.

C is now the closure of the original item set.

Closure(I)

```
procedure Closure( I an item set of the grammar G )
returns an item set of G, which may be the same as I

  Recall that P is the set G′s production rules; A,B ∈ N,
  the set of G′s non-terminals, and α,β ∈ (N ∪ Σ$)*
  are sequences of grammar symbols.

  let C be a copy of I
  repeat (
    foreach ( A → α•Bβ in C ) do (
      if ( B → π ∈ P and B → •π ∉ C) then
        add B → •π to C
      )
    )
  ) until ( C is unchanged )

  return C
```

GoTo of an Item Set and Grammar Symbol — Prose

Given an item set I and grammar symbol $\mathcal{X} \in N \bigcup \Sigma_{\$}$

1. Copy all the items from I that have $\mathcal{X}$ to the right of $\bullet$ into a new item set, call it K
2. Move the $\bullet$ marker past $\mathcal{X}$ for all the items in K

Use $Closure$ of K to complete the calculation of $GoTo(I, \mathcal{X})$:

$$GoTo(I, \mathcal{X}) = Closure(K)$$

GoTo(I, X)

```
procedure GoTo( I, X ) where
I is an item set of grammar G, and X ∈ N ⋃ Σ$
(a symbol of the grammar G or $)
GoTo returns an item set generated by the items in
I progressed past X, the result may be the same as I

   let K = { k ∈ I | X is to the right of • in k }
   let K′ = { k ∈ K | with • progressed past X }
   return Closure( K′ )
```

Generate SLR Action Table for Grammar G — Prose

An SLR Action Table ("Parsing Table") T has a row for each item set I_i of the grammar and a column for each grammar symbol **(less the starting goal)** and $. The table has three types of entries

SHIFT (AND GO TO) Shifts next element on the input queue to the stack (remembering the current parsing state i along with it) and goes to a new parsing state.

REDUCE (WITH RULE) Reduce the top elements on the stack with a particular production rule, pushing the resulting non-terminal back onto the top of the input queue. Assume the parsing state remembered with the element on the top of the stack.

REDUCE AND ACCEPT Reduce the top elements on the stack with the grammar **goal symbol's** singular production rule and **accept** the input as a valid sentence of the grammar.

Generate SLR Action Table for Grammar G — Prose

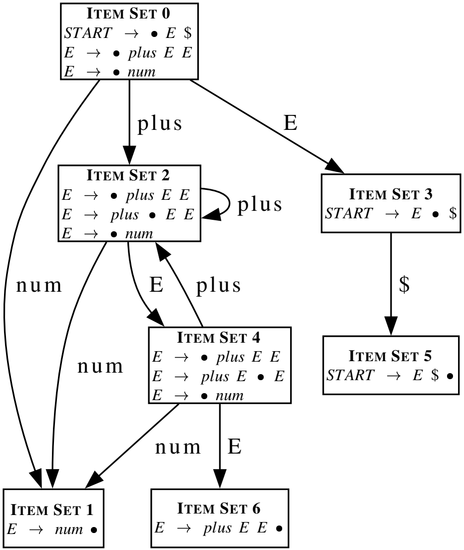
For each item set I_i, construct its Action Table row as follows:

1. For all items like $A \to \alpha \bullet \mathcal{X} \beta$ in I_i, make $T[i][\mathcal{X}]$ a SHIFT AND GO TO $GoTo(i, \mathcal{X})$ operation.
2. For all items like $A \to \alpha \bullet$ in I_i make the $T[i][t]$ operation REDUCE WITH $A \to \alpha$ for all terminals t in $FollowSet(A)$.[1]
3. For all items like $A \to \bullet \lambda$ in I_i make the $T[i][t]$ operation REDUCE WITH $A \to \lambda$ for all terminals t in $FollowSet(A)$.[1]

For the item set j that has item $S \to \pi \$ \bullet$, make the whole row $T[j][\cdot]$ the operation REDUCE AND ACCEPT.

[1] Raise a **CONFLICT ERROR** if $T[i][t]$ is already specified.

One more example: the Prefix Sum SLR Table



#	Rules
1	$START \rightarrow E\,\$$
2	$E \rightarrow plus\,E\,E$
3	$E \rightarrow num$

$$Follow(START) = \emptyset$$
$$Follow(E) = \{\$, num, plus\}$$

	num	*plus*	$	*E*
0	sh-1	sh-2		sh-3
1	r-3	r-3	r-3	
2	sh-1	sh-2		sh-4
3			sh-5	
4	sh-1	sh-2		sh-6
5	Reduce 1			
6	r-2	r-2	r-2	

Generate SLR Action Table — Setup

procedure SLRActionTable($\{I_i\}$ the item sets of G)
Where I_i is an item set (aka LR parser state) of the grammar G. Typically I_0 is the item set generated by the fresh start of the grammar **goal symbol** S which has one production rule terminated with $\$$.

Recall that $A, B \in N$, the set of G's non-terminals, $a \in \Sigma$ and $\alpha, \beta \in (N \bigcup \Sigma_\$)*$ are sequences of grammar symbols.

let $T[\cdot][\cdot]$ be the SLR action table with row i representing parser state i and columns for $X \in N \bigcup \Sigma_\$$ the grammar symbols of G augmented by $\$$.

Continued on next slide . . .

Generate SLR Action Table — Logic

. . . continued from last slide.

foreach (I_i item set in $\{I_i\}$) **do** (
  foreach ($\mathcal{X} \in N \bigcup \Sigma_\$$) **do** (
    if ($A \to \alpha \bullet \mathcal{X}\pi \in I_i$) **then** (
      set $T[i][\mathcal{X}]$ operation to SHIFT AND GO TO $GoTo(I_i, \mathcal{X})$
)
)
  foreach ($P = A \to \alpha \bullet | A \to \bullet \lambda \in I_i$) **do** (
    foreach ($f \in followSet(A)$) **do** (
      if ($T[i][f]$ is already specified) FAIL WITH CONFLICT
      set $T[i][f]$ operation to REDUCE WITH P
)
)
  if ($S \to \pi\$ \bullet \in I_i$) **then** (
    set $T[i][\cdot]$ operation to REDUCE WITH $S \to \pi\$$ AND ACCEPT
)
)