Syntactic Analysis

Ambiguity and Error Recovery

Ambiguous Languages

- A Context-Free Language (CFL) is inherently ambiguous if *every* Context-Free Grammar that generates the language is ambiguous
- However most ambiguous grammars found in practice are for unambiguous languages
  - Would like to make it unambiguous

Ambiguous Grammars

- If you have an ambiguous grammar for an unambiguous language, you can:
  - Write an unambiguous grammar
  - Use precedence and associativity to resolve parsing action conflicts

Example

<E> → <E> + <E> | <E> * <E> | (<E>) | id

Example

<E> → <E> + <E> | <E> * <E> | (<E>) | id

- Writing an unambiguous grammar

Example

<E> → <E> + <E> | <E> * <E> | (<E>) | id

- Writing an uambiguous grammar

<E> → <E> + <T> | <T>
<T> → <T> * <F> | <F>
<F> → (<E>) | id

<S> → <E> $
<E> → <E> + <E> | <E> * <E> | (<E>) | id

<S> → <E> $
<E> → <E> + <E> | <E> * <E> | (<E>) | id

s0
<S> → • <E>
<E> → • <E> + <E>
<E> → • <E> * <E>
<E> → • (<E>)
<E> → • id

s1
<S> → <E> •
<E> → <E> • + <E>
<E> → <E> • * <E>

s2
<S> → (• <E>)
<E> → • <E> + <E>
<E> → • <E> * <E>
<E> → • (<E>)
<E> → • id

s3
<E> → id •

s4
<S> → <E> + • <E>
<E> → • <E> + <E>
<E> → • <E> * <E>
<E> → • (<E>)
<E> → • id

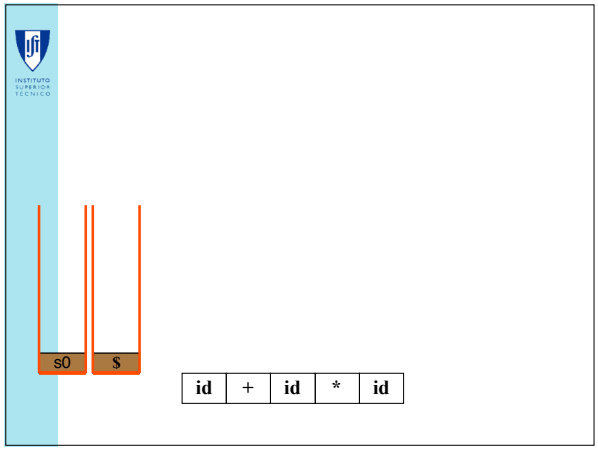
s5
<S> → <E> * • <E>
<E> → • <E> + <E>
<E> → • <E> * <E>
<E> → • (<E>)
<E> → • id

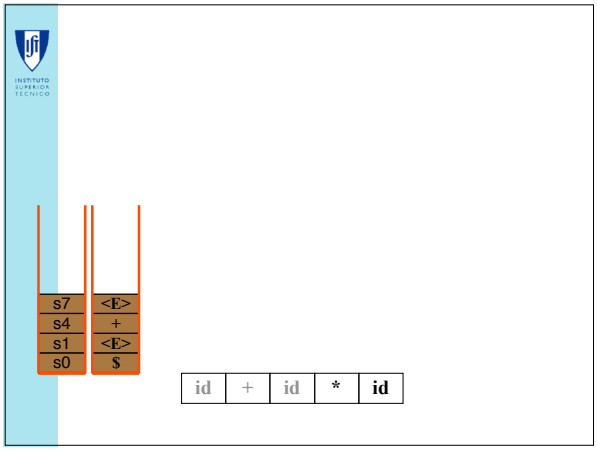
s6
<E> → <E> • + <E>
<E> → <E> • * <E>
<E> → (<E> •)

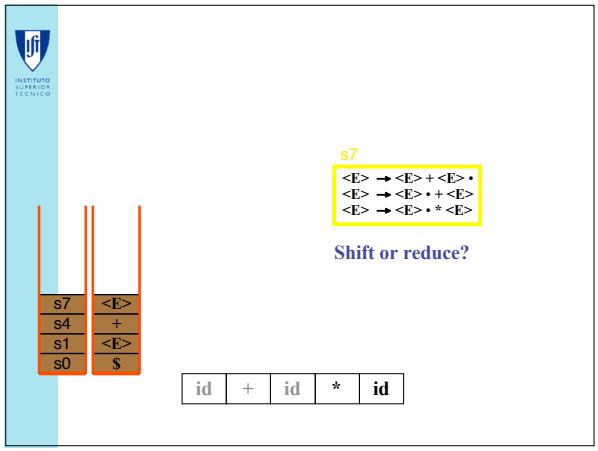
s7
<E> → <E> + <E> •
<E> → <E> • + <E>
<E> → <E> • * <E>

s8
<S> → <E> * <E> •
<E> → <E> • + <E>
<E> → <E> • * <E>

s9
<E> → (<E>) •







Using Precedence and Associativity

- Build the DFA and construct a parse table
- When there is a conflict use the precedence and associativity information

Another Example

<stmt> → if <expr> then <stmt> else <stmt>
<stmt> → if <expr> then <stmt>
<stmt> → other

Another Example

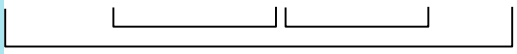
<stmt> → if <expr> then <stmt> else <stmt>
<stmt> → if <expr> then <stmt>
<stmt> → other

if ... then if ... then … else if ... then … else …

Another Example

<stmt> → if <expr> then <stmt> else <stmt>
<stmt> → if <expr> then <stmt>
<stmt> → other

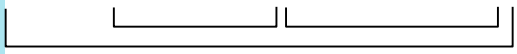
if ... then if ... then … else if ... then … else …



Another Example

<stmt> → if <expr> then <stmt> else <stmt>
<stmt> → if <expr> then <stmt>
<stmt> → other

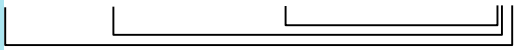
if ... then if ... then … else if ... then … else …



Another Example

<stmt> → if <expr> then <stmt> else <stmt>
<stmt> → if <expr> then <stmt>
<stmt> → other

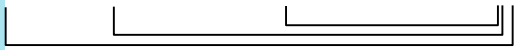
if ... then if ... then … else if ... then … else …



Another Example

<stmt> → if <expr> then <stmt> else <stmt>
<stmt> → if <expr> then <stmt>
<stmt> → other

if ... then if ... then … else if ... then … else …



Left associate, this is what we want

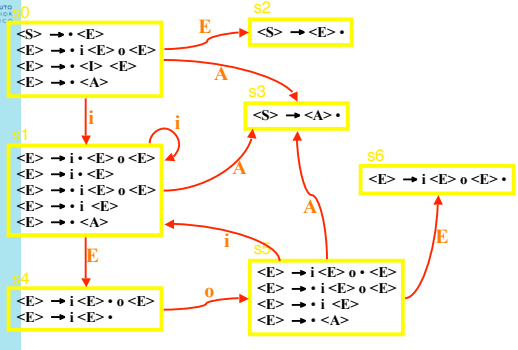
Another Example

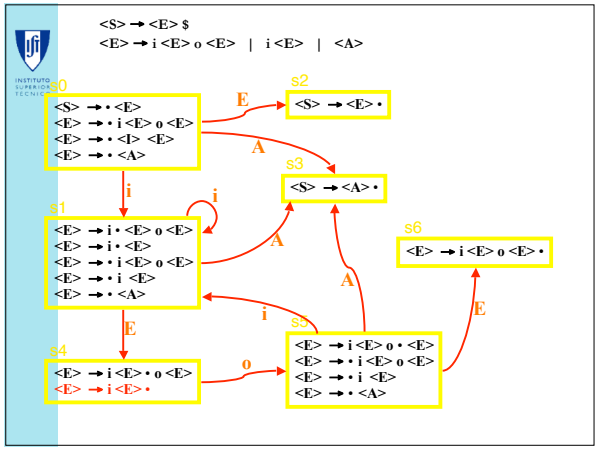
<stmt> → if <expr> then <stmt> else <stmt>
<stmt> → if <expr> then <stmt>
<stmt> → other

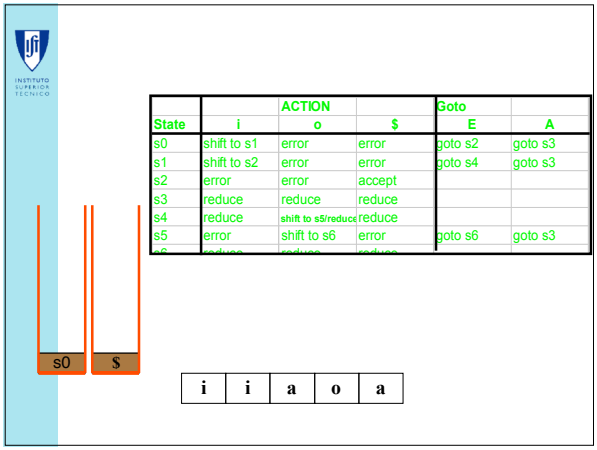
<S> → <E> $
<E> → i <E> o <E>
<E> → i <E>
<E> → <A>

Another Example

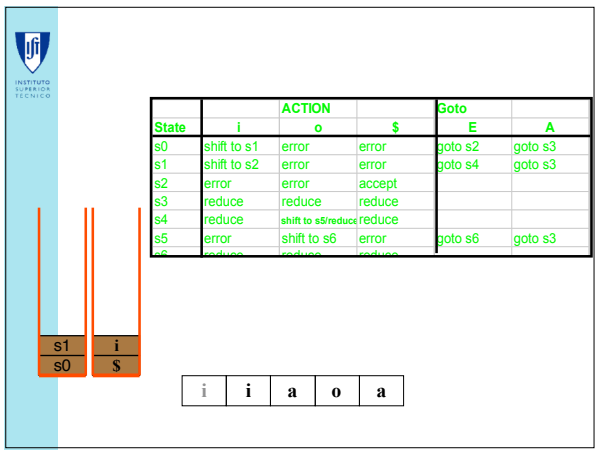
<S> → <E> $
<E> → i <E> o <E> | i <E> | <A>



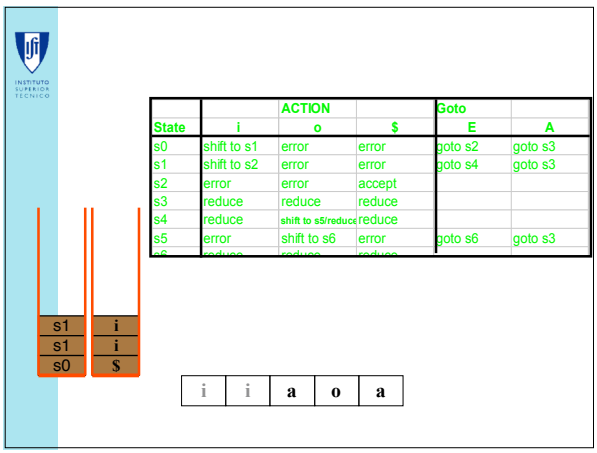




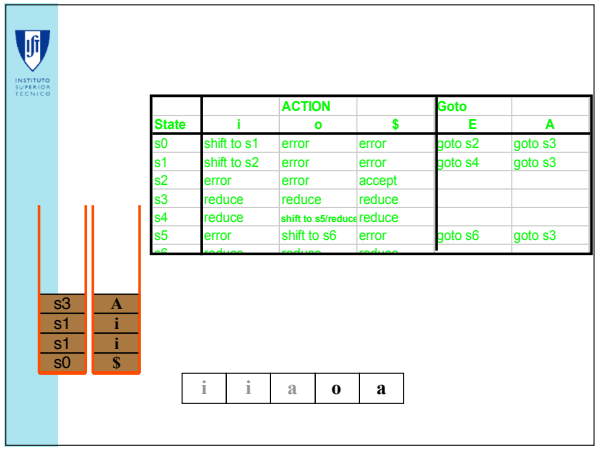
State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	shift to s5/reduce	reduce		
s5	error	shift to s6	error	goto s6	goto s3

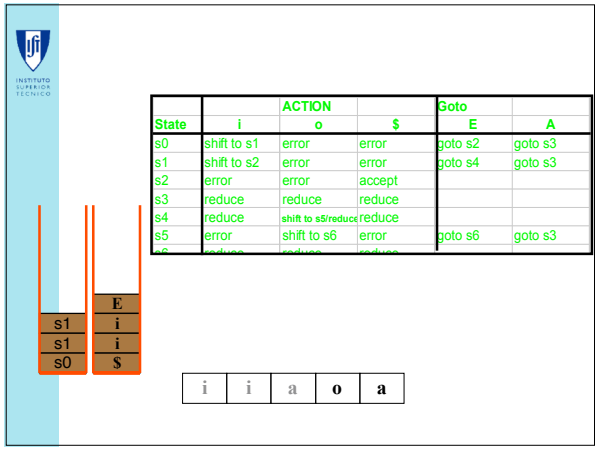


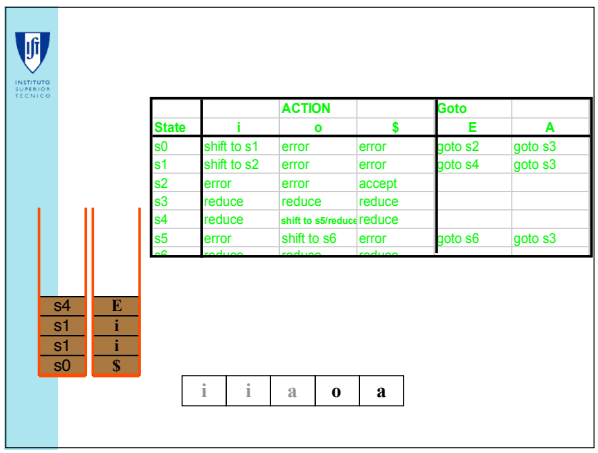
State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	shift to s5/reduce	reduce		
s5	error	shift to s6	error	goto s6	goto s3

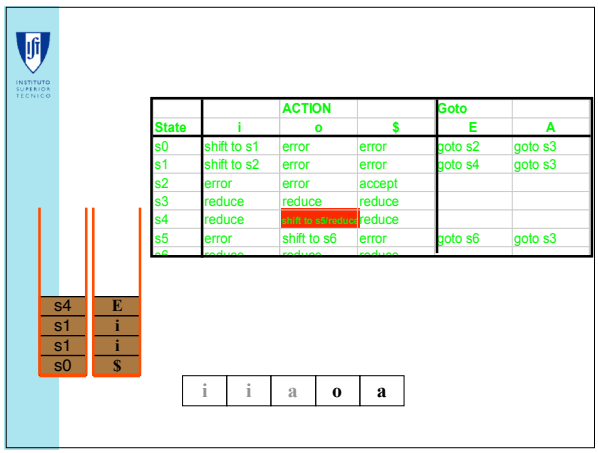


State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	shift to s5/reduce	reduce		
s5	error	shift to s6	error	goto s6	goto s3

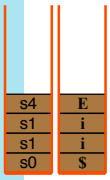

INSTITUTO SUPERIOR TECNICO
ACTION
Goto
State i o $ E A
s0 shift to s1 error error goto s2 goto s3
s1 shift to s2 error error goto s4 goto s3
s2 error error accept
s3 reduce reduce reduce
s4 reduce shift to s5/reduce reduce
s5 error shift to s6 error goto s6 goto s3
s3 A
s1 i
s1 i
s0 $
i i a o a


INSTITUTO SUPERIOR TECNICO
ACTION
Goto
State i o $ E A
s0 shift to s1 error error goto s2 goto s3
s1 shift to s2 error error goto s4 goto s3
s2 error error accept
s3 reduce reduce reduce
s4 reduce shift to s5/reduce reduce
s5 error shift to s6 error goto s6 goto s3
s1 E
s1 i
s1 i
s0 $
i i a o a

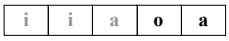

INSTITUTO SUPERIOR TECNICO
ACTION
Goto
State i o $ E A
s0 shift to s1 error error goto s2 goto s3
s1 shift to s2 error error goto s4 goto s3
s2 error error accept
s3 reduce reduce reduce
s4 reduce shift to s5/reduce reduce
s5 error shift to s6 error goto s6 goto s3
s4 E
s1 i
s1 i
s0 $
i i a o a


INSTITUTO SUPERIOR TECNICO
ACTION
Goto
State i o $ E A
s0 shift to s1 error error goto s2 goto s3
s1 shift to s2 error error goto s4 goto s3
s2 error error accept
s3 reduce reduce reduce
s4 reduce shift to s5/reduce reduce
s5 error shift to s6 error goto s6 goto s3
s4 E
s1 i
s1 i
s0 $
i i a o a

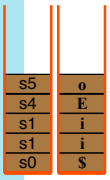
State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	[illegible]	reduce		
s5	error	shift to s6	error	goto s6	goto s3

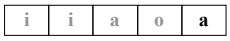


Since left associative, do the shift

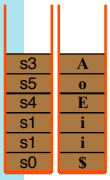


State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	shift to s5/reduce	reduce		
s5	error	shift to s6	error	goto s6	goto s3

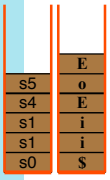




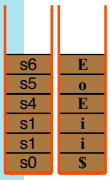
State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	shift to s5/reduce	reduce		
s5	error	shift to s6	error	goto s6	goto s3

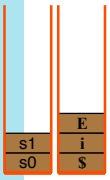


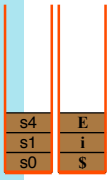
State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	shift to s5/reduce	reduce		
s5	error	shift to s6	error	goto s6	goto s3

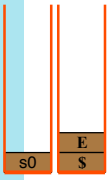


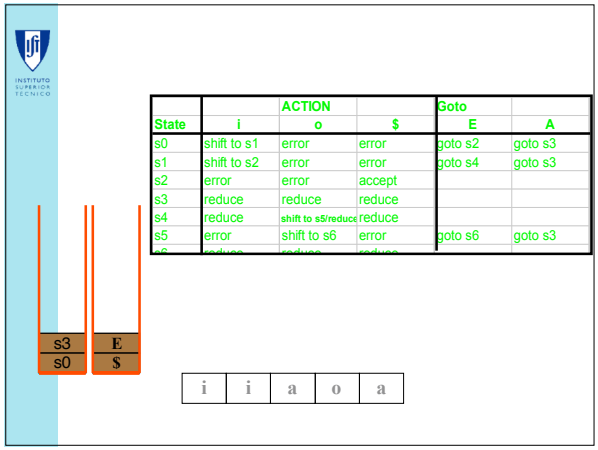
State	i	ACTION o	$	Goto E	A
s0	shift to s1	error	error	goto s2	goto s3
s1	shift to s2	error	error	goto s4	goto s3
s2	error	error	accept		
s3	reduce	reduce	reduce		
s4	reduce	shift to s5/reduce	reduce		
s5	error	shift to s6	error	goto s6	goto s3
s6	reduce	reduce	reduce		











Error Handling

- Programs are not always correct!!
- The Compiler needs to:
  - Report the presence of errors clearly and accurately
  - Recover from each error quickly enough to be able to detect subsequent errors
  - Try to avoid false error messages

Types of errors

- Lexical
- Syntactic
- Semantic
- Logical

Lexical Errors

- An error that produce a wrong token
- Possible lexical errors
  - misspelling of an identifier, keyword or operator

Syntactic Errors

- A program that does not satisfy the CFG of the language
- Examples
  - arithmetic expression with unbalanced parentheses
  - missing semicolon

Semantic Errors

- An error that needs context sensitive information to identify
- Examples
  - Operator applied to an incompatible operand
  - Accessing an undeclared variable

Logical Errors

- Errors in the Execution Model
- Examples
  - Infinitely recursive call
  - Accessing an array out of bounds
  - Dereferencing a null pointer

Types of Syntax Error Recovery

- Panic mode recovery
- Parse level recovery
- Error productions
- Global correction

Panic Mode Recovery

- On Discovering an Error
  - Pop zero or more states/symbols off the stack
  - Discard zero or more input symbols
  - Until we reach a point where parsing can continue
- User Defined Panic Non-terminals
  - Example: right bracket, semicolon

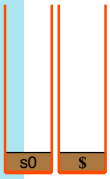
Panic Mode Recovery

- On Error
  - Pops stack till it reaches a state X that can perform a goto transition on one or more panic non-terminals
    - parsing is terminated if none found
  - Discards tokens from the input buffer till it finds a synchronizing token
    - A token that belongs to follow(A) of the panic non-terminal A encountered in the previous step
  - Pushes A and the state goto(S,A) on the stacks and resume parsing
- Same as pretending having seen A
- Issue: Which A and terminals to Choose?
  - Typical choices; statements and ';' or "end" tokens to synchronize

Example of panic mode error recovery

Grammar:

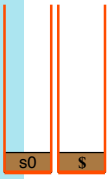
<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>



Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

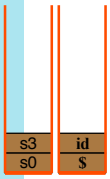


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

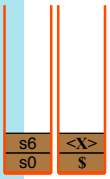


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

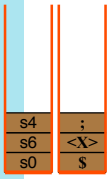


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

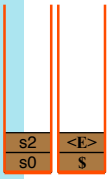


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

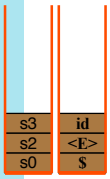


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

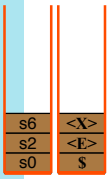


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

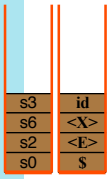


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

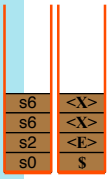


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

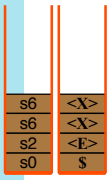


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>



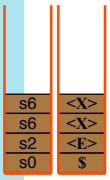
PANIC

id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>



PANIC

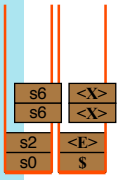
Panic Nonterminals = { **<E>** ... }
follow(<E>) = { ; , $}

id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>



PANIC

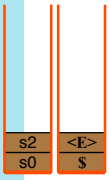
Panic Nonterminals = { **<E>** ... }
follow(<E>) = { ; , = }

id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>



PANIC

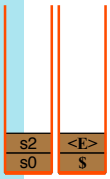
Panic Nonterminals = { **<E>** ... }
follow(<E>) = { ; , $}

id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>



PANIC

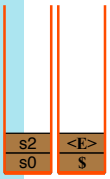
Panic Nonterminals = { **<E>** ... }
follow(<X>) = { ; , $ }

id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>



PANIC

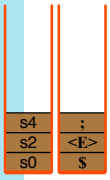
Panic Nonterminals = { **<E>** ... }
follow(<E>) = { ; , $ }

id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

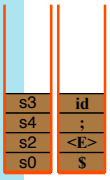


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

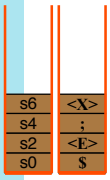


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

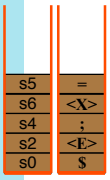


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

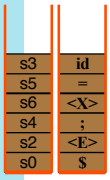


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

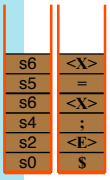


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

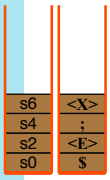


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

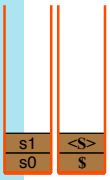


id	;	id	id	id	;	id	=	id

Example of panic mode error recovery

Grammar:

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X>

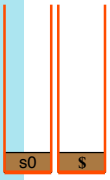


id	;	id	id	id	;	id	=	id

Parse Level Error Recovery

- On Error
  - Invoke a special routine to change the prefix of the remaining input tokens
  - and continue parsing
- Example
  - Replace a comma by a semicolon

Example of parser level error recovery



(	(	)	(	$

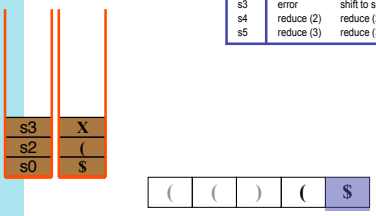
Parse Table

State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	

Example of parser level error recovery

Parse Table

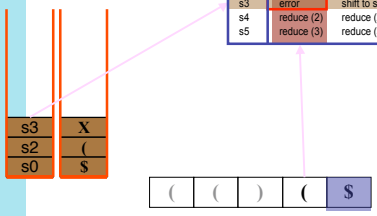
State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	



Example of parser level error recovery

Parse Table

State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	

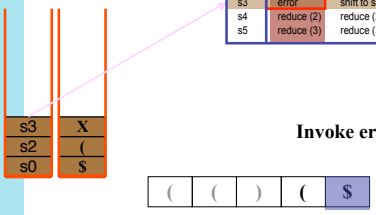


Example of parser level error recovery

Parse Table

State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	

Invoke error recovery routine

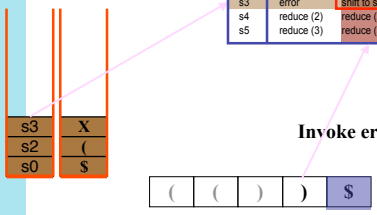


Example of parser level error recovery

Parse Table

State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	

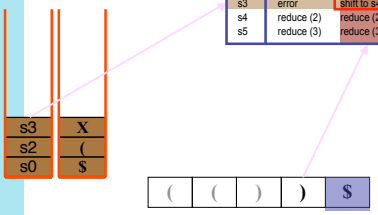
Invoke error recovery routine



Example of parser level error recovery

Parse Table

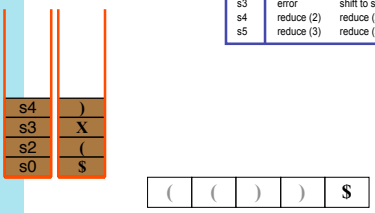
State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	



Example of parser level error recovery

Parse Table

State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	



Error Productions

- Add additional special productions of the form **A → α error β** to handle errors.
  - **error** is treated as a special terminal symbol
- On Error
  - Insert the **error** terminal as the first input token
  - Pop stack till it reaches a state **E** that can perform a goto transition on the **error** terminal
  - Parser shifts on the error terminal, the current state **F** = goto(**E**, **error**)
  - Discard the tokens from the input buffer till it finds a token for which a legal parse action can be performed from the state **F**

Example of Error Productions

- **Grammar**

  <S> → <E> $

  <E> → <E> ; <X> | <X> ;

  <X> → id | <X> = <X>

Example of Error Productions

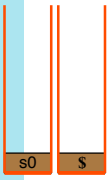
- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

Example of Error Productions

- Grammar

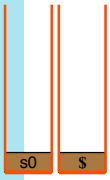
<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

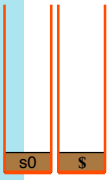


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

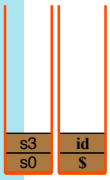


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

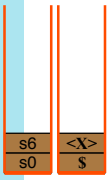


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

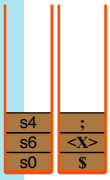


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

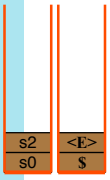


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

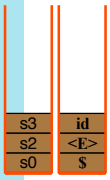


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

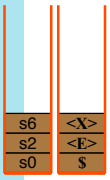


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

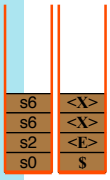


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

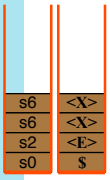


id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



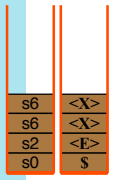
Error

id	;	id	id	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



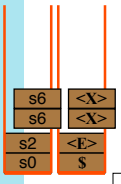
Error

id	;	id	id	error	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



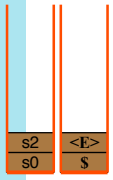
Error

id	;	id	id	error	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



Error

id	;	id	id	error	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

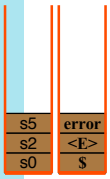
Error

id	;	id	id	error	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



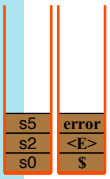
Error

id	;	id	id	error	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



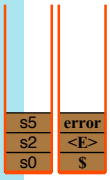
Error

id	;	id	id	error	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error

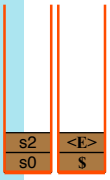


id	;	id	id	error	id	;	id	=	id

Example of Error Productions

- Grammar

<S> → <E> $
<E> → <E> ; <X> | <X> ;
<X> → id | <X> = <X> | error



id	;	id	id	error	id	;	id	=	id

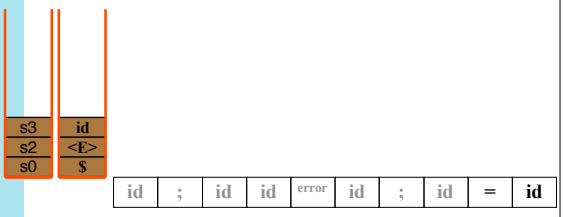
Example of Error Productions

- Grammar

  <S> → <E> $

  <E> → <E> ; <X> | <X> ;

  <X> → id | <X> = <X> | error



Global Correction

- Try to anticipate the programmer's action
- Make the program legal by choosing a minimal amount of changes
- Many Drawbacks
  - Costly
  - The changes might create a semantically correct program, but not the one the programmer intended to write!!!