DEPARTMENT OF COMPUTER SCIENCE
CARLOS III UNIVERSITY OF MADRID

Computer Science
Language Processors

Rules

- The duration of the test is **2.0 hours**
- Questions will not be answered during the test
- One cannot re-enter the classroom after leaving it
- The answers must be written using a pen (not a pencil)

Construct an LR (0) parser to evaluate sentences of the language corresponding to the following grammar:

```
7. S ::= A

8. S ::= A S

9. A ::= id = E

10.E ::= id

11.E ::= id + E

12.
```

5. **Calculate the canonical collection of configuration sets LR (0).**
6. **Represent the Automaton corresponding to the previous collection.**
7. **Obtain the parser tables LR (0). Indicates the problems they present.**
8. **Indicate the changes required on the previous table to obtain an SLR parser.**

The FIRST and FOLLOW sets for the non-terminal symbols of the grammar are:

Non-Terminal	FIRST	FOLLOW
S	Id	$
A	id	id $
E	Id	Id $

David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber

Solution:

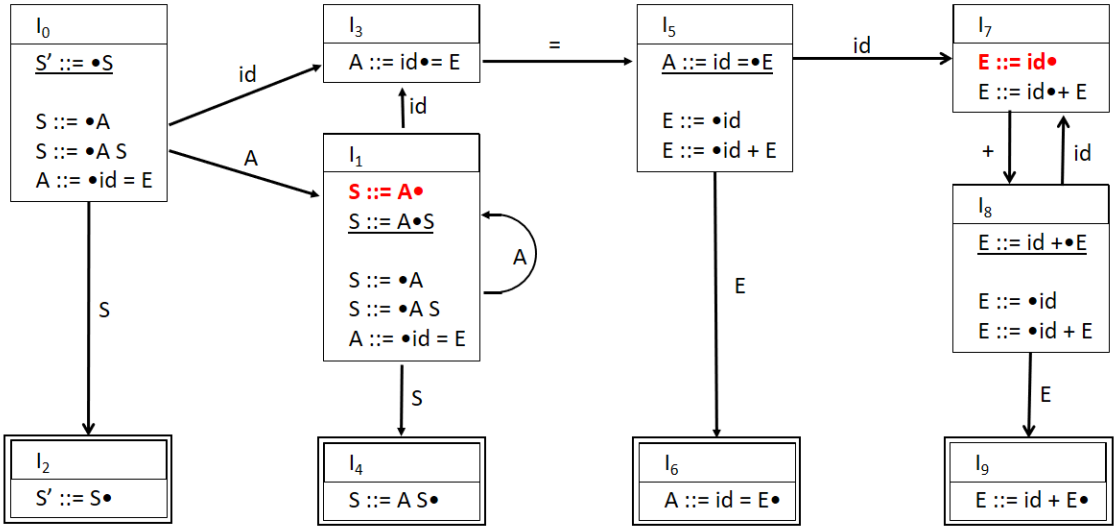
Augmented grammar::

```
0. S' ::= S
1. S ::= A
2. S ::= A S
3. A ::= id = E
4. E ::= id
5. E ::= id + E
```

To construct the set of configurations, we start from I_0, in which we have S' ::= •S. We will add the equivalent items to obtain I_0.

I_0
S' ::= •S
S ::= •A
S ::= •A S
A ::= •id = E

The corresponding automaton is:



The **LR(0)** table is (conflicts in red):

	+	=	id	$	S	A	E
0			S3		2	1	
1	R1	R1	**R1/S3**	R1	4	1	
2				Acc			
3		S5					
4	R2	R2	R2	R2			
5			S7				6
6	R3	R3	R3	R3			
7	**R4/S8**	R4	R4	R4			
8			S7				9
9	R5	R5	R5	R5			

Para determinar las Reducciones en SLR, nos basamos en los *token* determinados por *Siguiente* del NT reducido.

	FIRST	FOLLOW
S	id	$
A	id	$ id
E	id	$ id

The SLR table solves the two conflicts for this grammar:

	+	=	id	$	S	A	E
0			S3		2	1	
1			S3	R1	4	1	
2				Acc			
3		S5					
4				R2			
5			S7				6
6			R3	R3			
7	S8		R4	R4			
8			S7				9
9			R5	R5			

⇨ Es decir:

R1	S ::= A	$
R2	S ::= A S	$
R3	A ::= id = E	$ id
R4	E ::= id	$ id
R5	E ::= id + E	$ id

David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber

shift with A to I_0	shift with S to I_0	shift with id to I_0	Shift with A to I_1
I_1	I_2	I_3	I_1 (recursive)
S ::= A• S ::= A•S S ::= •A S ::= •A S A ::= •id = E	S' ::= S•	A ::= id•= E	S ::= A• S ::= A•S S ::= •A S ::= •A S A ::= •id = E
shift with id to I_1	shift with S to I_1	shift with = to I_3	shift with E to I_3
I_3 (repetido)	I_4	I_5	I_6
A ::= id•= E	S ::= A•S	A ::= id =•E E ::= •id E ::= •id + E	A ::= id = E•
shift with id to I_5	shift with + to I_7	shift with id to I_8	shift with E to I_8
I_7	I_8	I_7 (repeated)	I_9
E ::= id• E ::= id•+ E	E ::= id +•E E ::= •id E ::= •id + E	E ::= id• E ::= id•+ E	E ::= id + E•

David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber