Lenguajes y Sistemas Informáticos para la resolución de problemas complejos

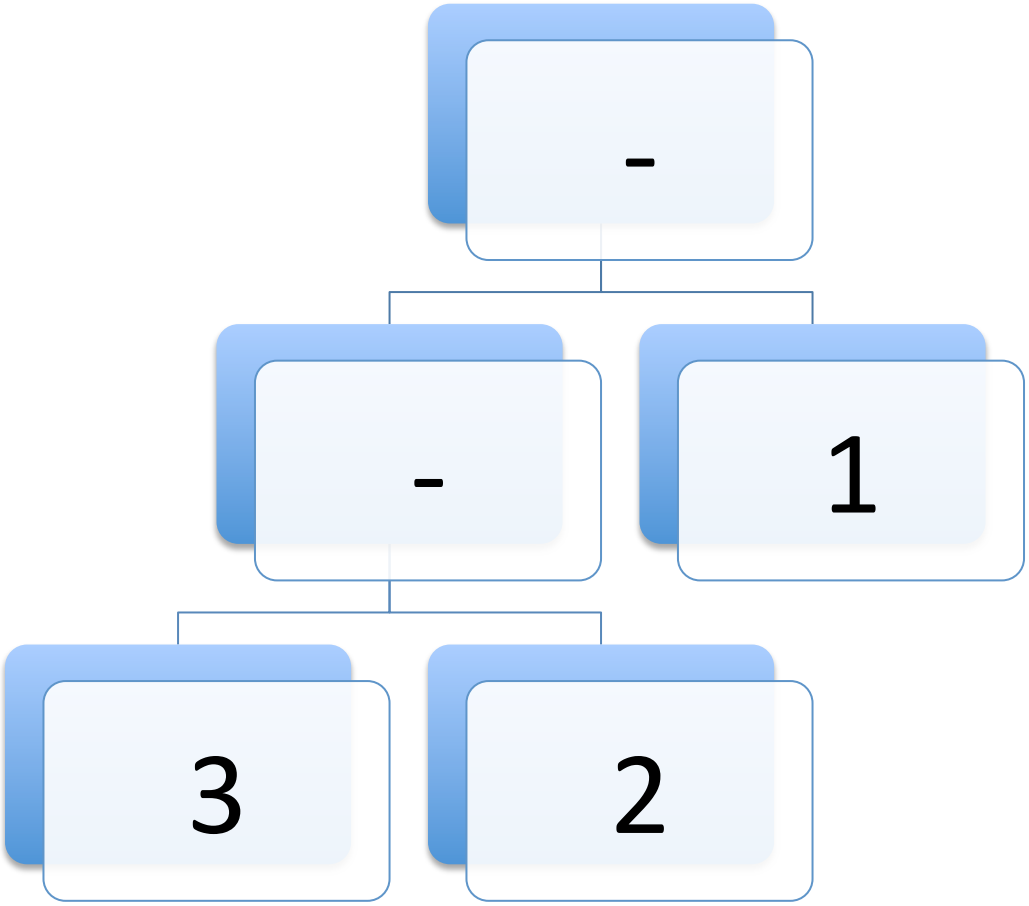
Procesadores de Lenguajes

Casiano Rodríguez León

3 - 2 - 1

Árbol Sintáctico Abstracto

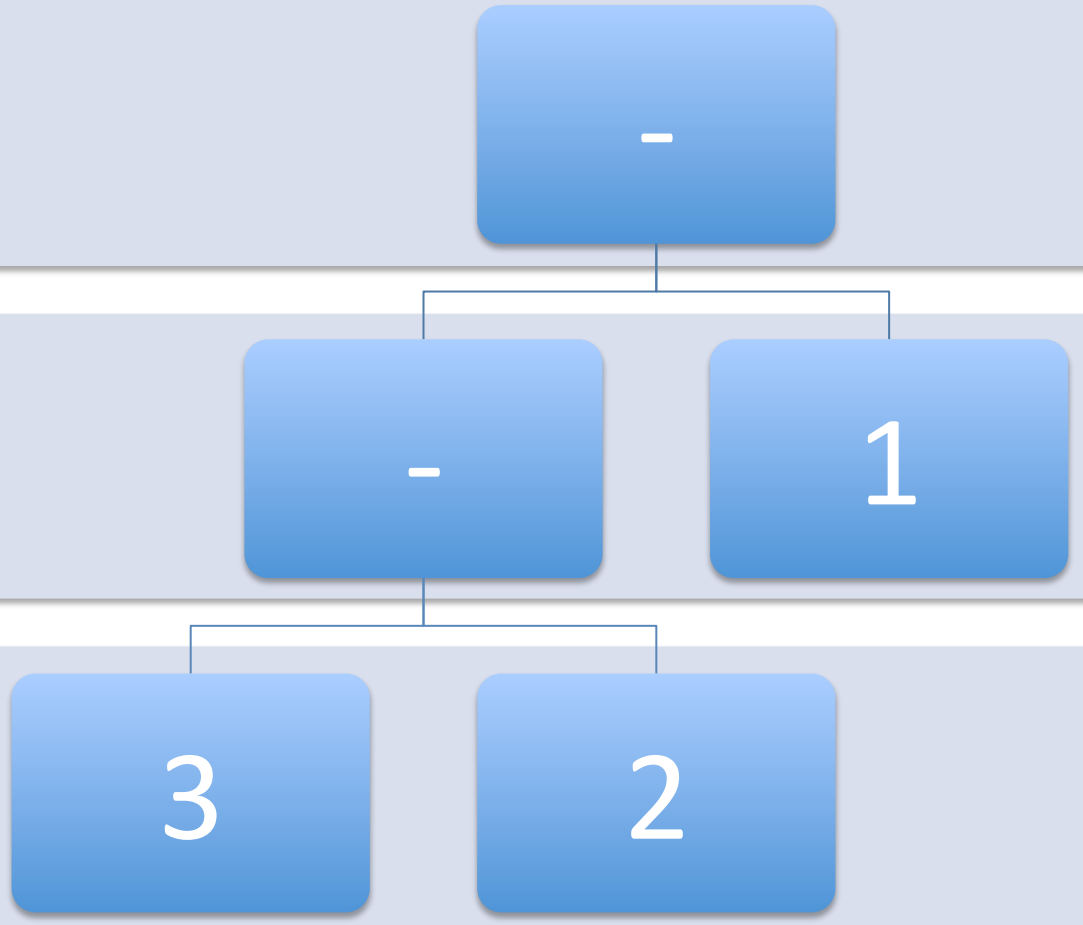
(3-2)-1



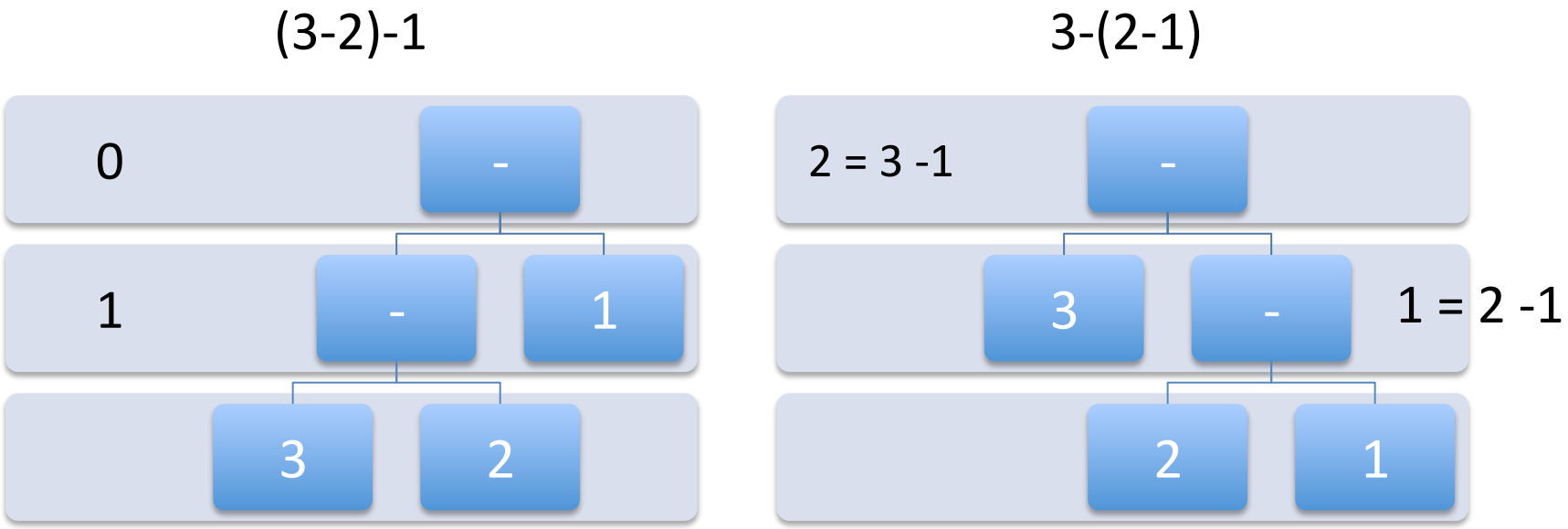
Semántica 3 - 2 - 1

0 = 1 - 1

1 = 3 -2



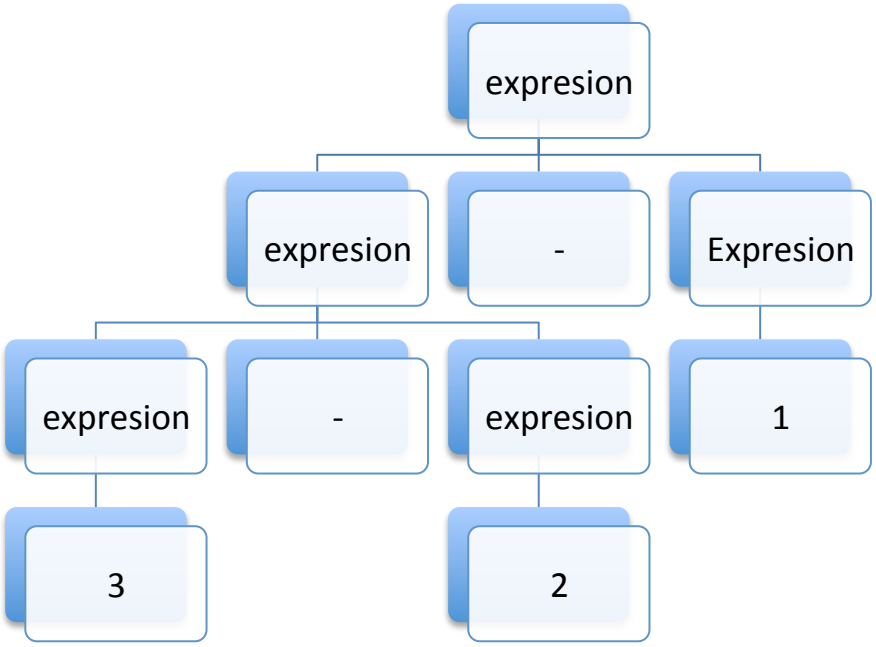
Semántica y Ambigüedad



Gramática Independiente del Contexto

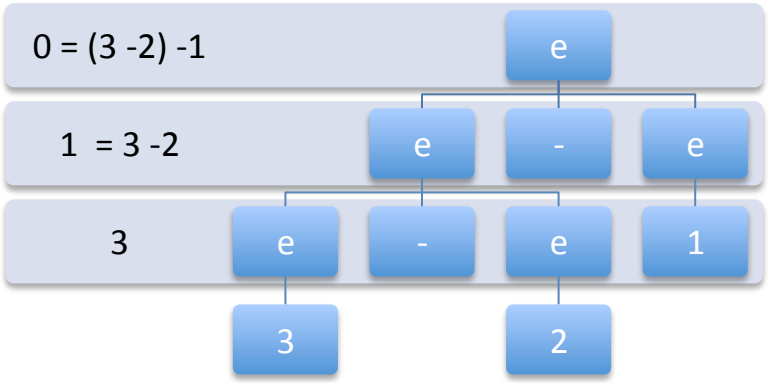
- expresion -> expresion ‘-’ expresion
- expresion -> NUMERO

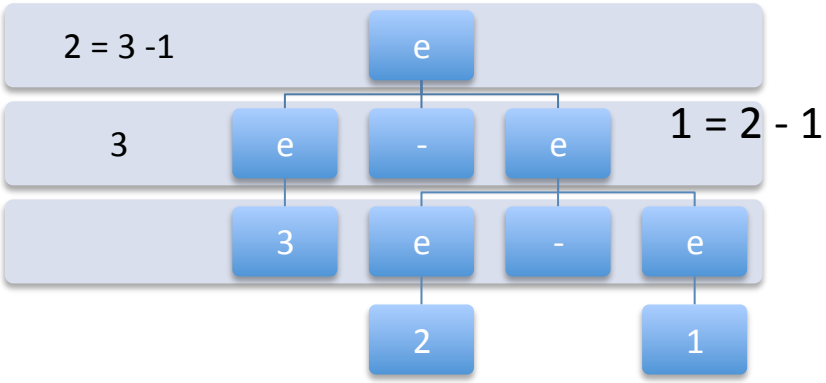
(3-2)-1



Gramática Ambigua

- expresion -> expresion ‘-’ expresion
- expresion -> NUMERO



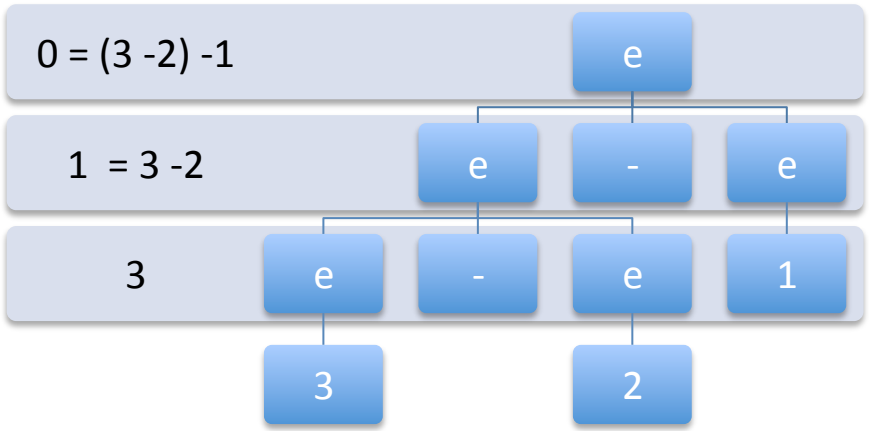


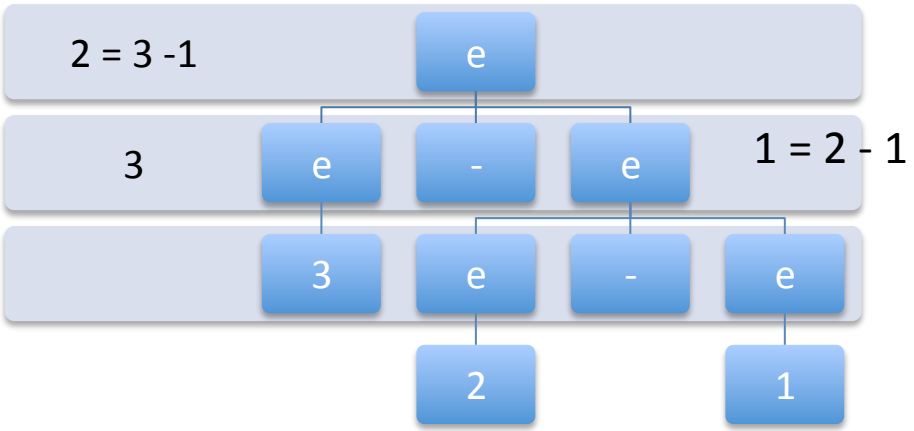
Esquema de Traducción (yacc)

e -> e '-' e { $$ = $1 - $3; }

e -> NUM { $$ = Number($1); }

3 – 2 - 1





Parsing: Construcción del Árbol

e -> e '-' e { $$ = $1 - $3; }

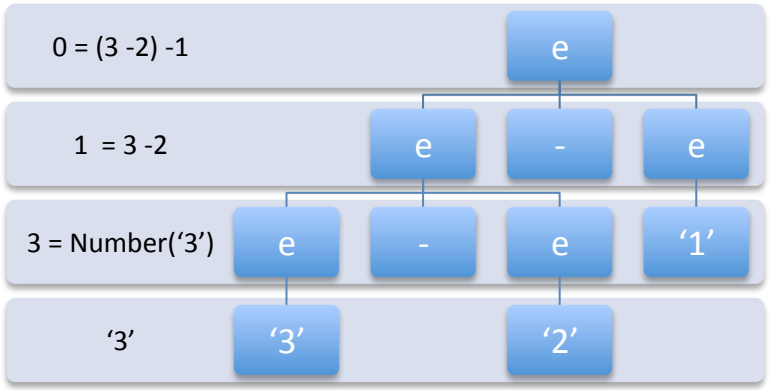
e -> NUM { $$ = Number($1); }

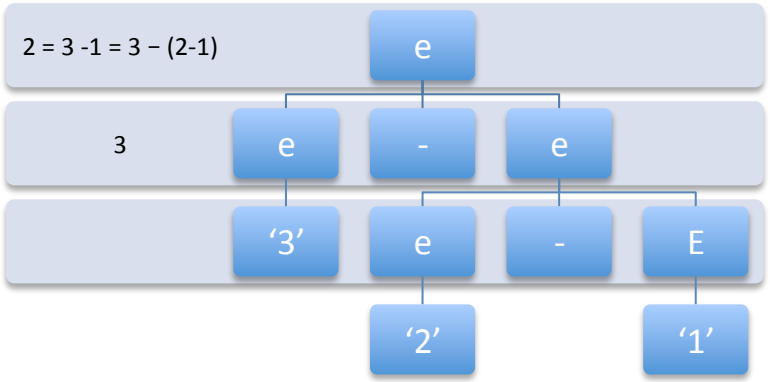
Análisis Sintáctico Ascendente:

.3 – 2 – 1 <= e. – 2 – 1 <= = e –. 2 – 1 <= = e – 2. – 1 <= e – e. – 1

¿Qué hacer?

1. <= e. – 1 <= e - . 1 <= e – 1. <= e – e. <= e.
2. <= e – e - . 1 <= e – e – 1. <= e – e – e. <= e – e. <= e.





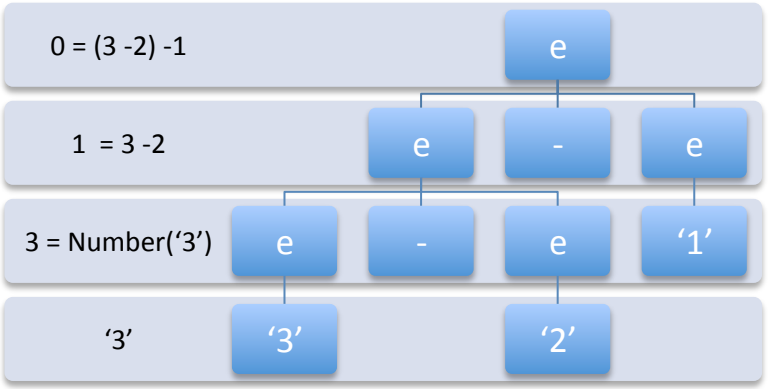
Conflicto Shift/Reduce

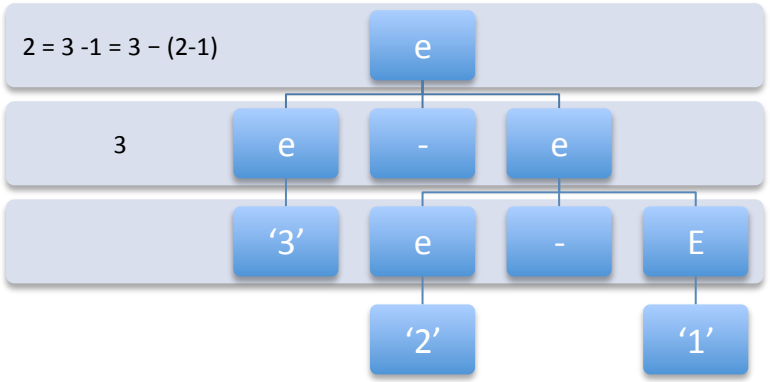
.3 – 2 – 1 <= e. – 2 – 1 <= = e –. 2 – 1 <= = e – 2. – 1 <= e – e. – 1

¿Qué hacer?

1. <= e. – 1 <= e - . 1 <= e – 1. <= e – e. <= e.
2. <= e – e - . 1 <= e – e – 1. <= e – e – e. <= e – e. <= e.

El conflicto puede verse como una lucha entre la regla e -> e '-' e y el terminal/token '-'





Un programa Yacc

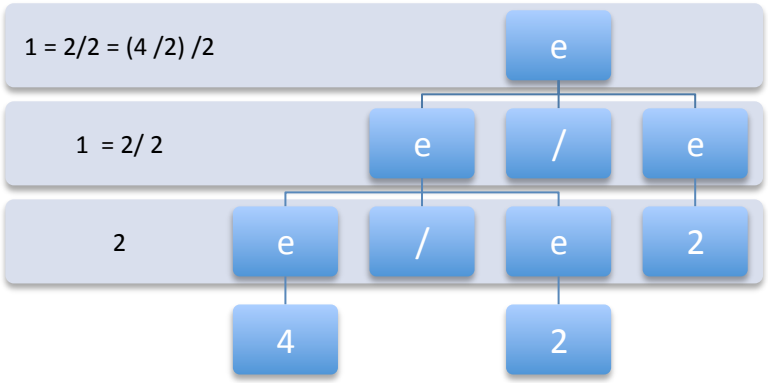
```
%left '-'
```

En la lucha entre la regla e -> e '-' e y el terminal/token '-' debe "ganar" la regla

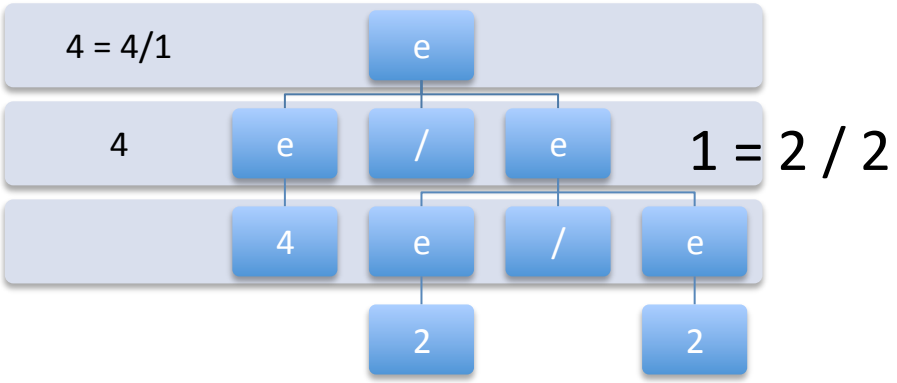
```
%%
s : e          { return $1; }
  ;

e : e '-' e { $$ = $1 - $3; }
  | NUM  { $$ = Number($1); }
  ;
```

Ambigüedad: Asociatividad
4/2/2

(4/2)/2

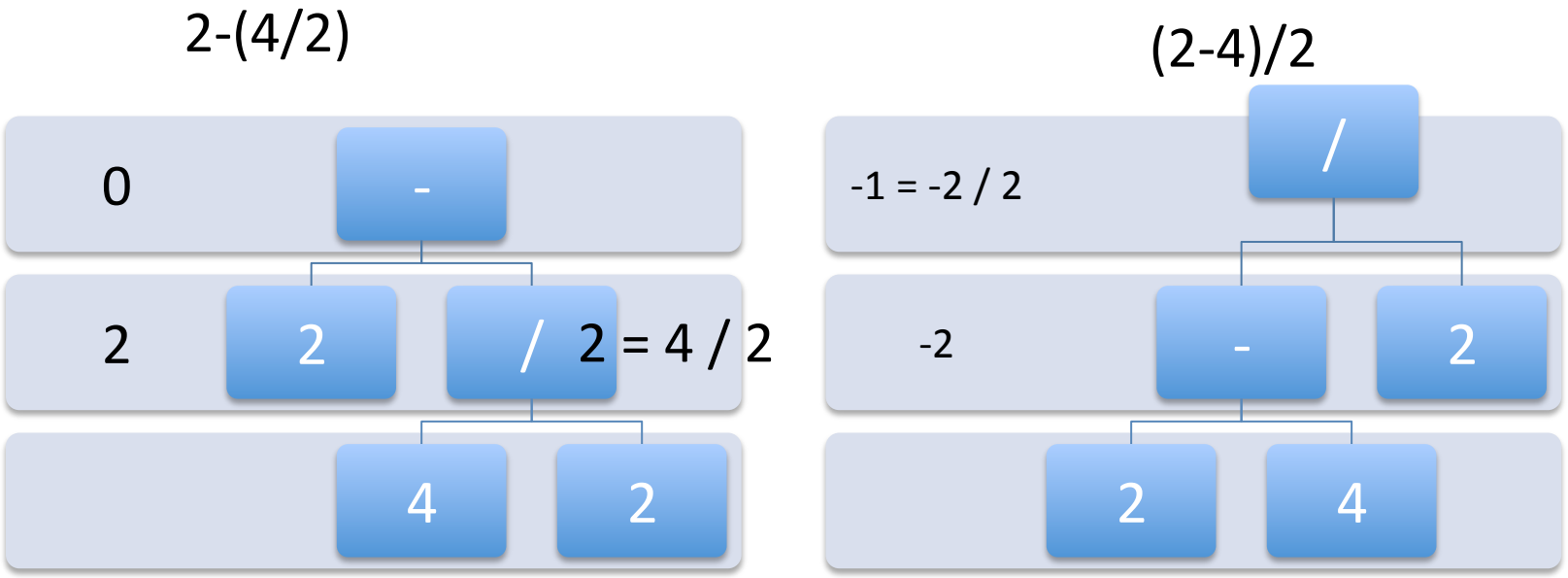


4/(2/2)



Ambigüedad: Prioridad

```
e : e '-' e { $$ = $1 - $3; }
  | e '/' e { $$ = $1 / $3; }
  | NUM { $$ = Number($1); }
  ;
```



Ambigüedad: Prioridad
2-4/2

2.-4/2 <= e.-4/2<=e-.4/2<=e-4./2<=e-e./2

¿Qué hacer?

1. <= e./2 <= e/. 2 <= e/e.<= e.
2. <= e-e/.2 <= e–e/2.<= e–e/e.<= e–e. <= e.

El conflicto es entre la regla e -> e ‘-’ e y el terminal ‘/’

Ambigüedad: Prioridad

Mas prioridad

```
%left '-'
%left '/'
%%
e : e '-' e { $$ = $1 - $3; }
  | e '/' e { $$ = $1 / $3; }
  | NUM { $$ = Number($1); }
  ;
```

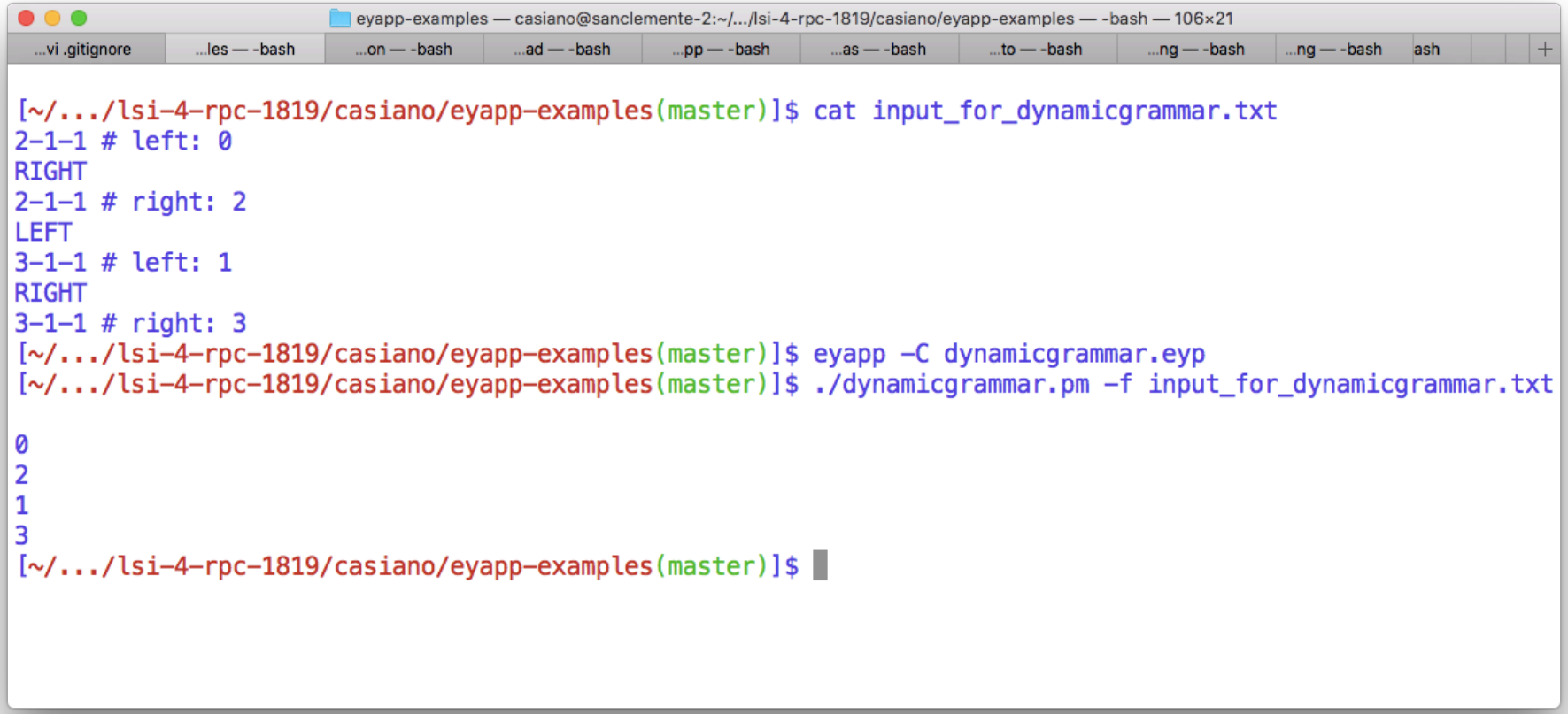
En la lucha entre reducir por la regla e -> e '-' e y desplazar el terminal '/' debe "ganar" el token

Dynamic Resolution of Shift-Reduce Conflicts

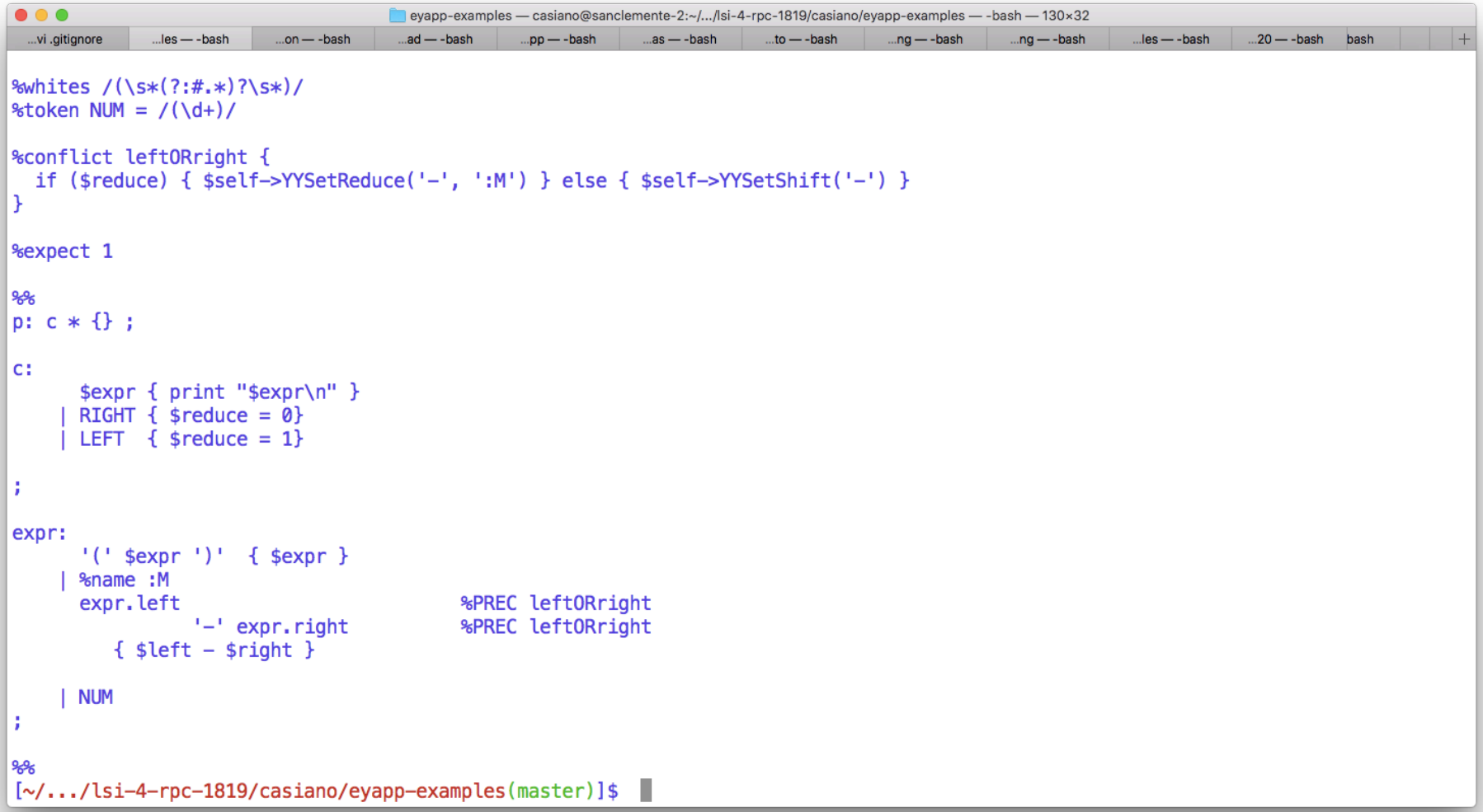
Write a language that accepts lists of two kind of commands: *arithmetic expressions* like *4-2-1* or one of two *commands*: *left* or *right*.

- When a *right* command is issued, the semantic of the '-' operator is changed to be right associative.
- When a *left* command is issued the semantic for '-' returns to left associative interpretation.

Dynamic Resolution of Shift-Reduce Conflicts



Dynamic Resolution of Shift-Reduce Conflicts



Recursos

- Repositorio GitHub con los recursos de la charla: https://github.com/ULL-LSI/campus-america-2019
- Apuntes de Procesadores de Lenguajes. Curso 2018/2019: https://ull-esit-pl-1819.github.io/introduccion/
- Rodriguez-Leon, Casiano & Garcia-Forte, L. (2011). Solving Difficult LR Parsing Conflicts by Postponing Them. Comput. Sci. Inf. Syst.. 8. 517-531. 10.2298/CSIS101116008R.
- Parse Eyapp en CPAN
- Parsing Strings and Trees with Parse::Eyapp (An Introduction to Compiler Construction). 2010