

Calculating *First* for the grammar:

```

Expr  -> Term Expr'
Expr' -> + Term Expr' | - Term Expr' | ε
Term  -> Factor Term'
Term' -> * Factor Term' | / Factor Term' | ε
Factor -> (Expr) | num | id

```

	num	id	+	-	*	/	()	ε
	num	id	+	-	*	/	()	ε

Rule	Expr	Expr'	Term	Term'	Factor
2		ε		ε	
3a		+, -		*, /	
3a					(, id, num
3a			(, id, num		
3a	(, id, num				