

## 2.3 Analizatzaile lexikoa implementatzeko pausoak: ADIBIDE BAT

- a) ESPEZIFIKAZIOA: Token-motak, adierazpen erregularrak, atributuak eta lotutako ekintzak zehaztu:

token-mota	adierazpen erreg.	atributuak	ekintza
identifikadore	$L(L D)^*$	Erref_S_T: Sinboloen taulako posizioa	Kate:=upcase(katea) Erref_S_T:= Gehitu(S_T, katea)
konstante osokoa	$D^+$	Balioa: osokoa	Balioa:= Zenbakira_Pasa(Katea)
banatzaile	$(zurig tab lerro\_buk)^+$		ez egin ezer
errore	$karaktere\_arraroa^+$		errorea tratatu

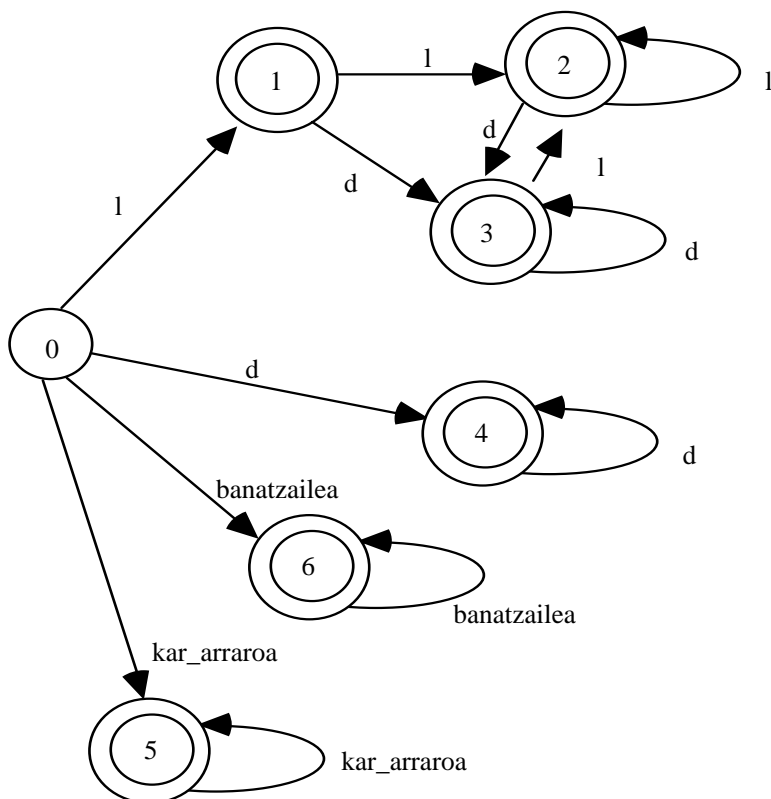
*Gehitu:*

*Sinboloen\_Taula x Katea --> Sinboloen\_Taula x S\_Tren\_erakuslea*

Identifikadorea ez baldin badago, gehitu egingo du eta bere posizioa bueltatuko du, bestela bere posizioa bueltatu

*Zenbakira\_Pasa: Katea --> osokoa*

- b) AFED edo AFD bat egin (klaseko algoritmoak erabiliz edo eskuz):



Token bat ezagutzeko algoritmoa:

## Algoritmoa Lortu\_tokena

## hasiera

```
e:= Hasierako_Egoera;
```

```
k:= Lehen_Karakterea();
```

**while** trantsiziorik\_dago (e, k) **loop**
$$e := \text{trantsizioa}(e, k);$$

```
k:= Hurrengo_Karakterea();
```

**end;****if** bukaerakoa\_da(e)

**then** tokena lortu da (atributuen kalkulua,

sinboloen taulan sartu, ...)

**else** kate erroreduna (detektatutako errore-mota,

ekintzak, ...)

**bukaera**

## ANALIZATZAILEA LEXIKOA: Token ez-esanguratsuen tratamendua

(oharrak, banatzaileak, karaktere arraroen kateak, ...):

**while** ez dago token esanguratsua **loop**

Lortu\_tokena

**end**