Create the LR0 item sets for the following set of productions. The tokens are all the words in boldface, and the symbols ‘;’, ‘:=’, and ‘,’.

**CFG Productions**

```
goal   →  decl ⊢
decl   →  constdecl
        →  vardecl
constdecl →  const id := expn
           →  const id : type := expn
vardecl →  var idlist := expn
           →  var idlist : type
idlist →  id
           →  id , idlist
type   →  int
        →  real
expn   →  number
```

**Item Sets**

1: * goal  →  ∧ decl ⊢  on decl to state 2
decl   →  ∧ constdecl  on constdecl to state 3
decl   →  ∧ vardecl  on vardecl to state 4
constdecl →  ∧ const id := expn  on const to state 5
           →  ∧ const id : type := expn  same
vardecl →  ∧ var idlist := expn  on var to state 6
vardecl →  ∧ var idlist : type  same

2: * goal  →  decl ∧ ⊢  on ⊢ to state 7

3: * decl  →  constdecl ∧

4: * decl  →  vardecl ∧

5: * constdecl →  const ∧ id := expn  on id to state 8
           * constdecl →  const ∧ id : type := expn  same

6: * vardecl →  var ∧ idlist := expn  on idlist to state 9
           * vardecl →  var ∧ idlist : type  it same
idlist →  ∧ id  on id to state 10
idlist →  ∧ id , idlist  same

7: * goal  →  decl ⊢ ∧

8: * constdecl →  const id ∧ := expn  on := to state 11
           * constdecl →  const id ∧ : type := expn  on : to state 12
9:  * vardecl → var idlist ∧ := expn  on := to state 13
     * vardecl → var idlist ∧ : type  on : to state 14
10:  * idlist → id ∧
     * idlist → id ∧ , idlist  on , to state 15
11:  * constdecl → const id := ∧ expn  on expn to state 16
     expn → ∧ number  on number to state 17
12:  * constdecl → const id : ∧ type := expn  on type to state 18
     type → ∧ int  on int to state 19
13:  * vardecl → var idlist := ∧ expn  on expn to state 21
     expn → ∧ number  on number to state 17
14:  * vardecl → var idlist : ∧ type  on type to state 22
     type → ∧ int  on int to state 19
     type → ∧ real  on real to state 20
15:  * idlist → id , ∧ idlist  on idlist to state 23
     idlist → ∧ id  on id to state 10
     idlist → ∧ id , idlist same
16:  * constdecl → const id := expn ∧
17:  * expn → number ∧
18:  * constdecl → const id : type ∧ := expn  on := to state 24
19:  * type → int ∧
20:  * type → real ∧
21:  * vardecl → var idlist := expn ∧
22:  * vardecl → var idlist : type ∧
23:  * idlist → id , idlist ∧
24:  * constdecl → const id : type := ∧ expn  on expn to state 25
     expn → ∧ number  on number to state 17
25:  * constdecl → const id : type := expn ∧