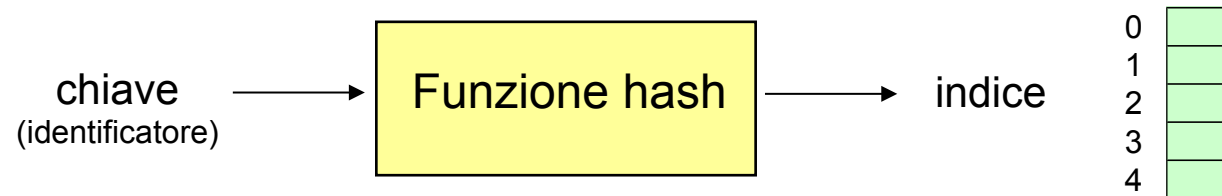
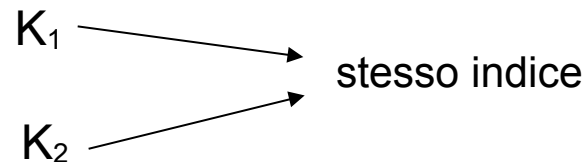


Hash Table

- HT = array $[0..TOT-1]$ of buckets



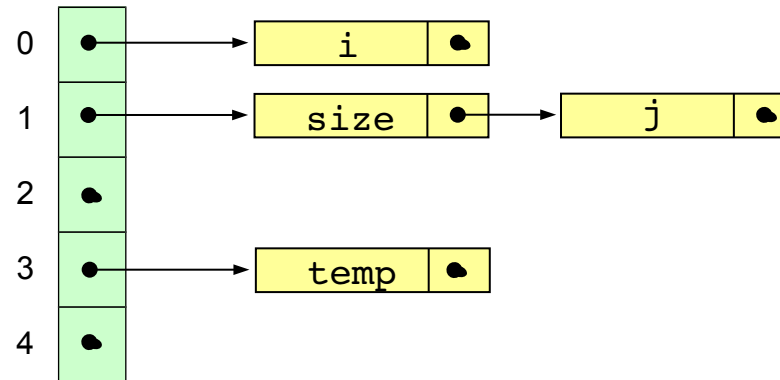
- Problemi di collisioni:



- Dimensione dell'array (**TOT**)
 - Tipicamente definito nel codice del compilatore (centinaia / migliaia)
 - Numero **primo** (distribuzione migliore)

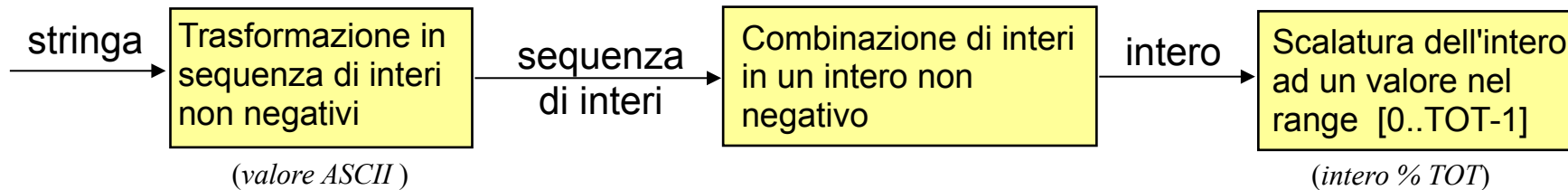
Risoluzione delle Collisioni

- Bucket = lista lineare \Rightarrow collisione \rightarrow inserimento in testa alla lista



Mapping

- Trasformazione: stringa di caratteri \rightarrow intero $\in [0..TOT-1]$ in 3 passi:



- Passo intermedio: uso di un "peso" α :

$$h = (\alpha^{n-1}c_1 + \alpha^{n-2}c_2 + \dots + \alpha c_{n-1} + c_n) \% TOT = \left(\sum_{i=1}^n \alpha^{n-i} c_i \right) \% TOT$$

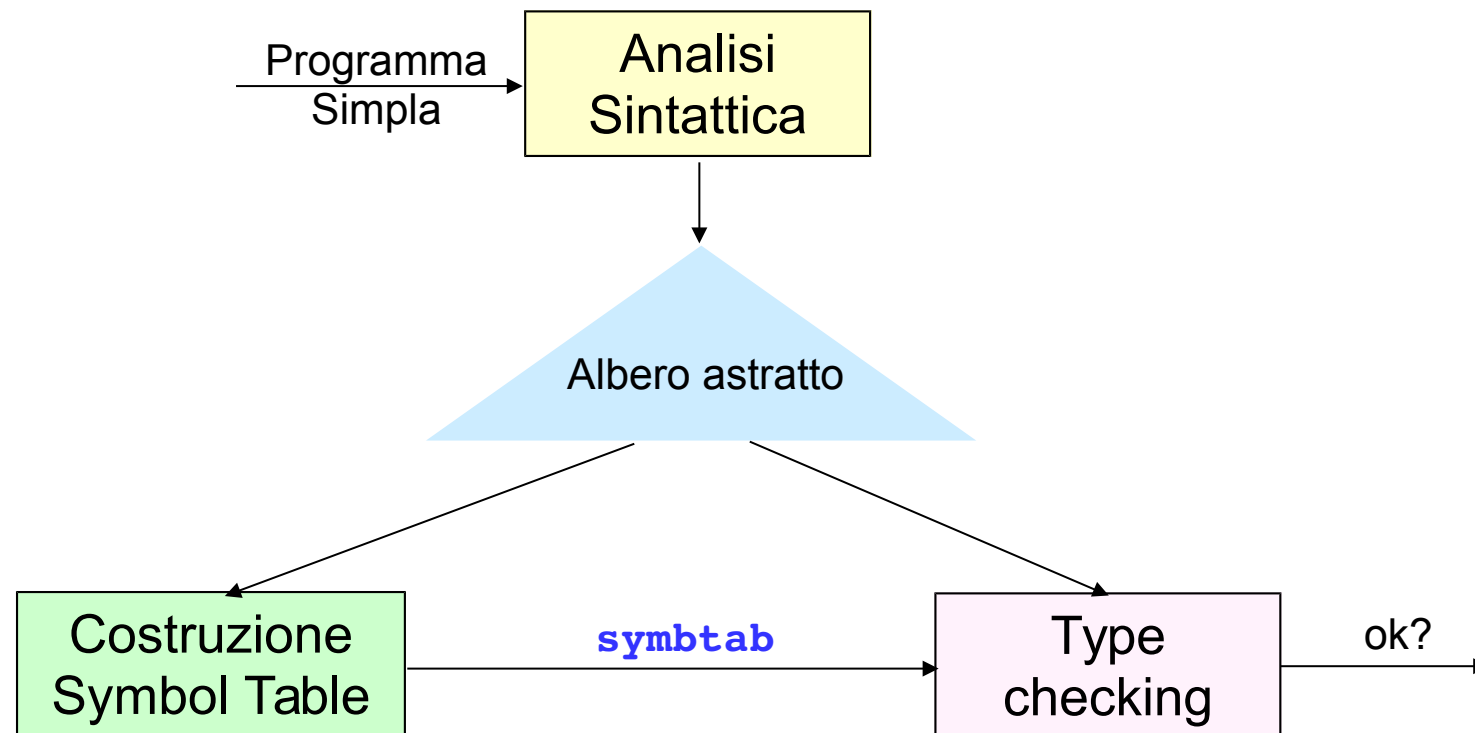
- Problema overflow \rightarrow inserimento della operazione di modulo nella somma (es: $\alpha = 16$)

```
#define TOT ...
#define SHIFT 4

int hash(char* id)
{
    int h=0;

    for(i=0; id[i] != '\0'; i++)
        h = ((h << SHIFT) + id[i]) % TOT;
    return h;
}
```

Approccio in Simpla



Symbol Table



- **nome:** puntatore al nome dell'identificatore
- **classe** $\in \{ \text{VAR}, \text{PAR}, \text{FUN} \}$
- **oid:** object identifier
 - VAR, PAR $\rightarrow \text{oid} \in [1, 2, \dots] : \text{numerazione } \underline{\text{relativa}}$ nell'ambiente
 - FUN $\rightarrow \text{oid} \in [1, 2, \dots] : \text{numerazione } \underline{\text{assoluta}}$ nel programma
- **tipo:** tipo dell'oggetto (INT, REAL, STRING, BOOL, VOID)
- **ambiente:** ambiente locale (per funzioni) = referenza alla tabella dei simboli locali
- **formali**
 - num = numero dei parametri formali
 - descr = vettore di puntatori a simboli (parametri formali del modulo) nell'ambiente
- **next:** puntatore al descrittore successivo (stesso indice hash)

Symbol Table (ii)

```
numero: integer;

func fattoriale(n: integer): integer
  fact: integer;
body
  if n == 0 then
    fact = 1;
  else
    fact = n * fattoriale(n-1);
  end;
  return fact;
end;

func stampaFattoriali(tot: integer): void
  i, f: integer;
body
  for i=0 to tot do
    f = fattoriale(i);
    writeln("Il fattoriale di ", i, "è ", f);
  end;
end;

body
  read(numero);
  if numero < 0 then
    writeln("Il numero ", numero, "non è valido");
  else
    stampaFattoriali(numero);
  end;
end.
```

Symbol Table (iii)

