Strutture dati dinamiche in C (II)

Laboratorio di Linguaggi di Programmazione
a.a. 2001/2002

dott.ssa Francesca A. Lisi
lisi@di.uniba.it

Sommario

- Le liste concatenate (ancora … ma in modo più formale)
- L'acquisizione di grammatiche (ancora ... ma con ulteriori suggerimenti)
- Introduzione agli alberi binari

Riferimenti

- cap. 12 di Deitel & Deitel
- par. 6.5 di Kernighan & Ritchie
- cap.1 di Semeraro

Dichiarazione di un tipo *lista*

```
typedef ... TipoElemLista;

struct StructLista {
  TipoElemLista info;
  /* elemento della lista */
  struct StructLista *next;
  /* puntatore all'elemento successivo */
};

typedef struct StructLista TipoNodoLista;
typedef TipoNodoLista *TipoLista;
```

Inizializzazione di una lista

```
void InitLista(TipoLista *l)
  /* Inizializza l alla lista vuota. */
{
  *l = NULL;
}

bool TestListaVuota(TipoLista l)
  /* Restituisce TRUE se l e` la lista
  vuota, FALSE altrimenti. */
{
  return (l == NULL);
}
```

Inserimento elementi in testa ad una lista

```
void InserisciTestaLista
  (TipoLista *l, TipoElemLista elem)
{
  TipoLista paux;

  paux = malloc(sizeof(TipoNodoLista));
  paux->info = elem;
  paux->next = *l;
  *l = paux;
}
```

Inserimento elementi in coda ad una lista

```
void InserisciCodaLista
  (TipoLista *l, TipoElemLista elem)
  /* Versione iterativa. */
{
  TipoLista ultimo;   /* puntatore usato
  per la scansione */
  TipoLista paux;

  /* creazione del nuovo nodo */
  paux = malloc(sizeof(TipoNodoLista));
  paux->info = elem;
  paux->next = NULL;
```

Inserimento elementi in coda ad una lista (II)

```
 /* scansione della lista */
  if (*l == NULL)
    *l = paux;
  else {
    ultimo = *l;
    while (ultimo->next != NULL)
      ultimo = ultimo->next;
     /* concatenazione del nuovo nodo */
    ultimo->next = paux;
  }
}  /* InserisciCodaLista */
```

Esercizi

- Scrivere una funzione C che restituisca l'elemento in testa ad una lista
- Scrivere una funzione C che restituisca l'elemento in coda ad una lista
- Scrivere una funzione C che risolva il problema dell'inserimento di un elemento in coda ad una lista in maniera *ricorsiva*
- Scrivere una funzione C che faccia la copia di una lista in ingresso
- Scrivere una funzione C che stampi una lista di interi

Esempio con una lista di caratteri

```
/* programma char-list.c */
#include <stdio.h>
#include <stdlib.h>

#define TRUE  1
#define FALSE 0
typedef int bool;
typedef char TipoElemLista;

#include "tipolis.c"
#include "liste.c"
```

Esempio con una lista di caratteri (II)

```
int main(void) {
  TipoLista l;

  l = malloc(sizeof(TipoNodoLista));
  l->info = 'B';  l->next = NULL;
  /* l punta alla lista ('B') */
  InserisciTestaLista(&l, 'A');
  /* l punta alla lista('A','B') */
  InserisciCodaLista(&l, 'C');
  /* l punta alla lista('A','B','C') */
  return 0;
}
```

Acquisizione di grammatiche:
un esempio illustrativo

Cfr. grammatica a pag. 5 del libro “Appunti di Teoria dei Linguaggi Formali” - G. Semeraro

1) <frase_nominale> ::= <parte_nominale><parte_verbale><parte_nominale>

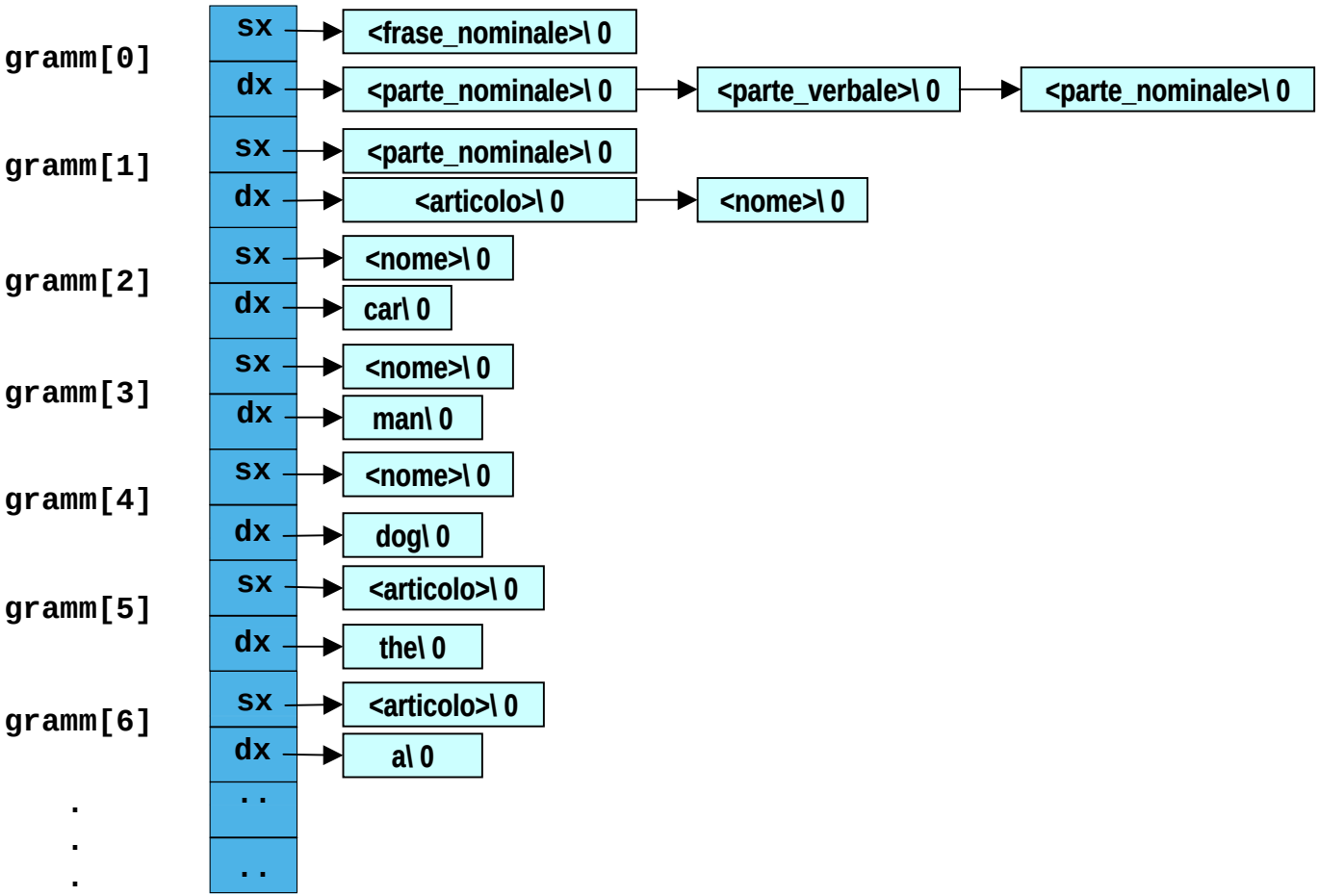
2) <parte_nominale> ::= <articolo><nome>

3) <nome> ::= car | man | dog

4) <articolo> ::= the | a

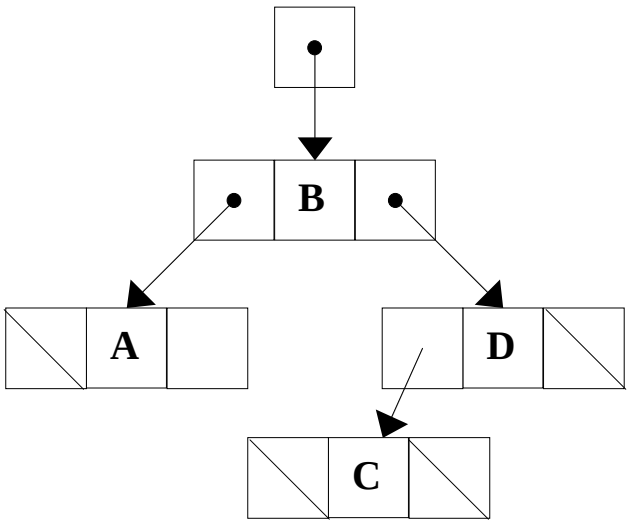
5) <articolo> ::= hits | eats

Acquisizione di grammatiche:
un esempio illustrativo (II)



Alberi binari

- Ogni elemento può puntare ad almeno due elementi della struttura
- In un albero binario, ogni nodo contiene due puntatori
  - Nessuno, uno, od entrambi dei quali possono essere **`NULL`**
  - Se un nodo *a* punta ad un nodo *b*, *a* sidice ***nodo padre*** e *b* si dice ***nodo figlio***
  - Il ***nodo radice*** è un nodo senza padre
  - Un nodo senza figli si dice ***nodo foglia***



Alberi binari (II)

```
struct el_alb_bin {
… /* dati */
struct el_alb_bin *figlio_sx;
struct el_alb_bin *figlio_dx;
} *alb_bin = null;

/* creazione ed inizializzazione di new_el */
struct el_alb_bin *new_el =
    (struct el_alb_bin *)malloc(sizeof(struct el_alb_bin));
(*new_el).figlio_sx = (*new_el).figlio_dx = null;

alb_bin = new_el;
```