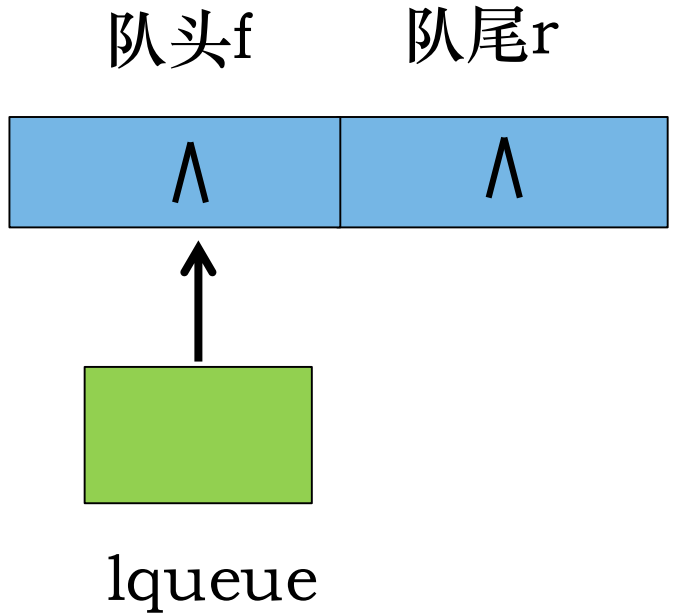
3.10 链队列

```
1   typedef int DataType;
2   struct Node
3   {
4           DataType data;
5           struct Node *link;
6   };
7   typedef struct Node *PNode;
8   struct Queue
9   {       PNode           f;
10          PNode           r;
11  };
12  typedef struct Queue *LinkQueue;
```

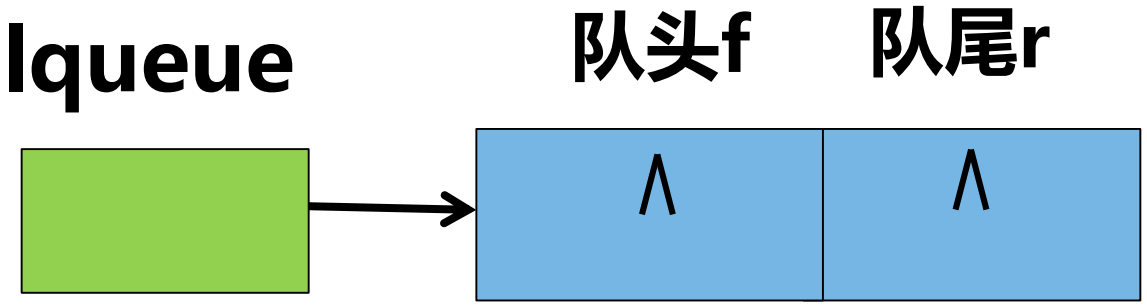
3.10.1 创建空队列

算法3-24

```
1   LinkQueue  SetNullQueue_Link()//创建空队列
2   {
3           LinkQueue lqueue;
4           lqueue = (LinkQueue)malloc(sizeof(struct Queue));
5           if (lqueue != NULL)
6           {
7                   lqueue->f = NULL;
8                   lqueue->r = NULL;
9           }
10          else
11                          printf("Alloc failure! \n");
12          return  lqueue;
13  }
```

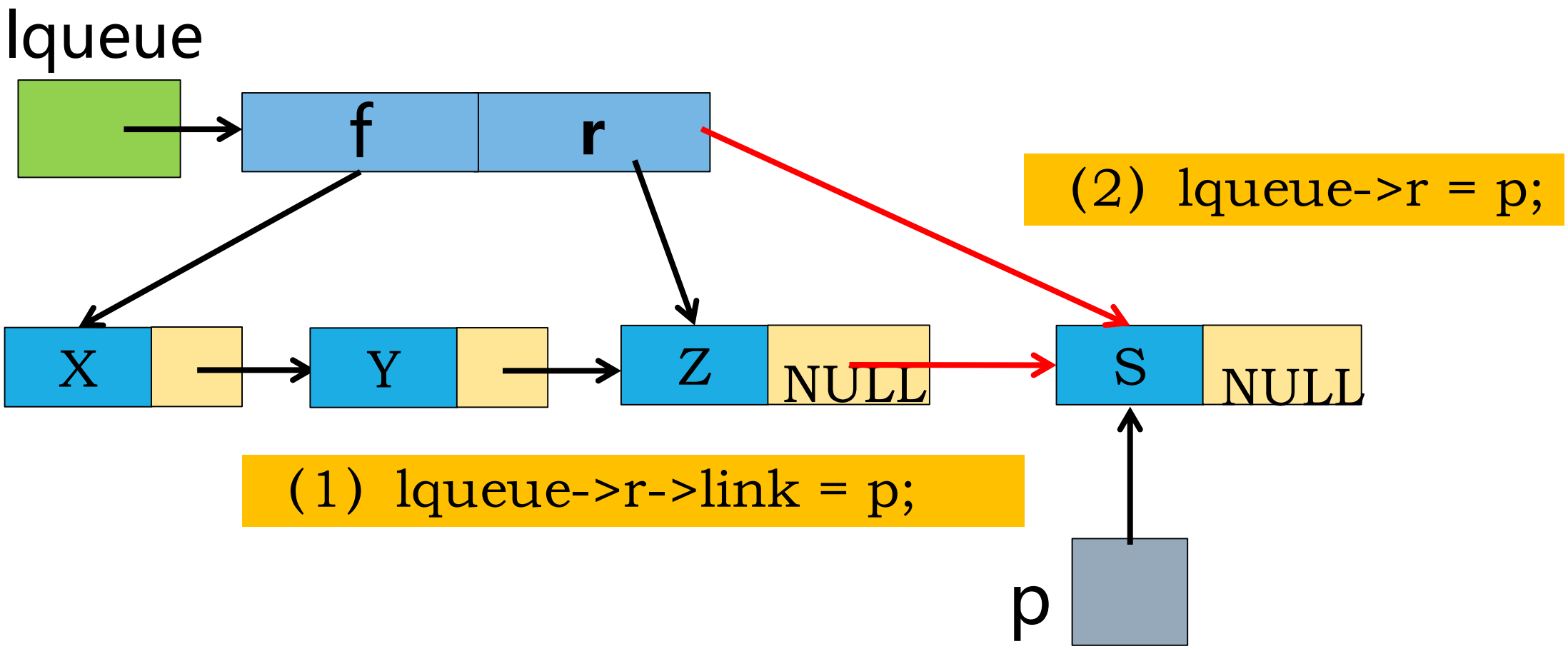


3.10.2 判断队列是否为空 算法3-25




```
1  //判断队列是否为空
2  int IsNullQueue_Link(LinkQueue lqueue)
3  {
4          return (lqueue->f == NULL);
   }
```

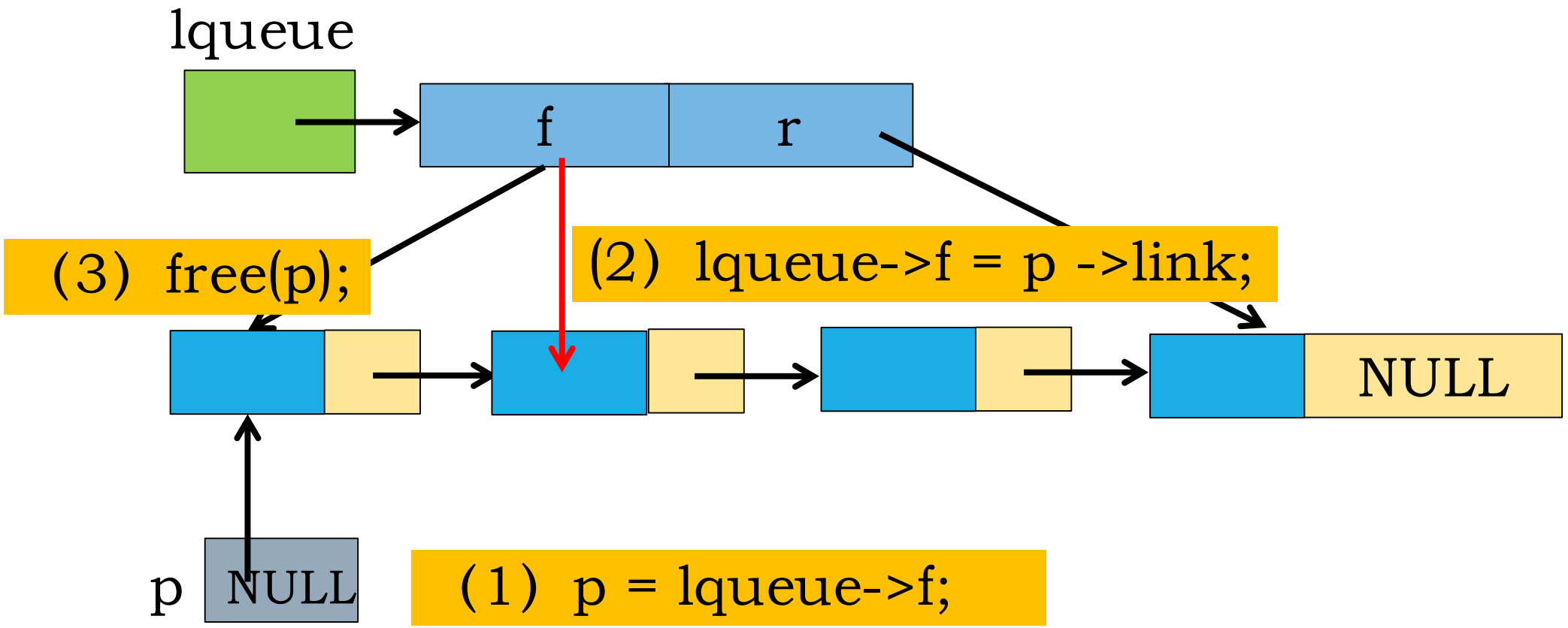
3.10.3 入队



```
1  void EnQueue_link(LinkQueue lqueue, DataType x){     //入队操作
2          PNode  p;
3          p = (PNode)malloc(sizeof(struct Node)); //申请结点空间
4          if (p == NULL)
5                  printf("Alloc failure!");
6          else{
7                          p->data = x;          //数据域赋值
8                          p->link = NULL;       //指针域赋值
9                          if (lqueue->f == NULL) {    //空队列的特殊处理
10                                 lqueue->f = p;
11                                 lqueue->r = p;
12                         }
13                         else{
14                                 lqueue->r->link = p; //插入队尾
15                                 lqueue->r = p;      //修改队尾指针
16                         }
17                 }
18 }
```

算法3-26

3.10.4 出队



3.10.4 出队

算法3-27

```
1   void DeQueue_link(LinkQueue lqueue) //出队
2   {
3           struct Node  * p;
4           if (lqueue->f == NULL)  //判断队列是否为空
5                   printf( "It is empty queue!\n ");
6           else
7           {
8                   p = lqueue->f;    //p指向队头结点，以方便后面的释放
9                   lqueue->f = lqueue->f->link;  //修改队头指针
11                  free(p);                      //是否结点空间
12          }
13  }
```