Sur la notion d'état (en informatique)

Emmanuel Beffara

9ème journée académique sur l'enseignement de l'informatique
21 mai 2025

Caractérisation de l’état dans un programme en Python

Un programme itératif simple

```
def log_2(n):
    acc = 1
    i = 0
    while acc < n:
        acc = acc * 2
        i = i + 1
    return i
```

Consigne

- Décrire l'exécution de l'appel `log_2(12)`
- Préciser ce qui fait partie de l'état

Un programme à récursion terminale

```python
def pgcd(a, b):
    if b == 0:
        return a
    else:
        return pgcd(b, a % b)
```

Consigne

- Décrire l'exécution de `pgcd(21, 28)`
- En déduire ce qui fait partie de l'état

Un programme en diviser pour régner

1. Compléter le code suivant :

```
def quicksort(tab, a, b):
    """
    Trie en place le tableau tab entre
    les indices a inclus et b exclu.
    """

    ...
```

   On suppose qu'on dispose d'une fonction `partition(tab, a, b)` qui partitionne `tab` entre `a` et `b` avec `tab[a]` comme pivot, en supposant `a < b`, et qui renvoie la position finale du pivot.

2. Décrire l'exécution sur un cas simple mais pas trop, par exemple :

```
tab = [6, 2, 9, 0, 1]
quicksort(tab, 0, 5)
```

3. Caractériser l'état d'une façon qui permette de justifier la description de l'exécution donnée précédemment

```python
def quicksort(tab, a, b):
    if a + 1 >= b: return
    m = partition(tab, a, b)
    quicksort(tab, a, m)
    quicksort(tab, m + 1, b)

def partition(tab, a, b):
    """ Partitionne tab entre a et b avec tab[a] comme pivot,
    en supposant a < b. Renvoie la position finale du pivot. """
    j = a + 1 # premier élément de la zone à trier
    k = b - 1 # dernier élément de la zone à trier
    while j <= k: #  tab[a + 1 : j] <= tab[a] < tab[k + 1 : b]
        if tab[j] <= tab[a]:
            j += 1
        elif tab[a] < tab[k]:
            k -= 1
        else: # tab[j] > tab[a] >= tab[k]
            tab[j], tab[k] = tab[k], tab[j]
            j += 1
            k -= 1
    tab[a], tab[k] = tab[k], tab[a]
    return k
```

L’état dans un protocole

Le protocole du bit alterné

Principe

On se situe à la couche 2 du modèle OSI, la couche liaison qui communique des trames. On considère une situation avec un émetteur et un récepteur identifiés.

- À chaque trame envoyée, l'émetteur ajoute un bit en fin de trame.
- Le récepteur répond par un accusé de réception terminé avec le bit inversé.
- Le bit utilisé dans l'accusé de réception est renvoyé lors de l'envoi suivant.

Associé à un délai d'attente, ce protocole simple permet de détecter la perte d'une trame ou la perte d'un accusé de réception.

Analyse

Décrire le protocole en explicitant l'état et ses transformations.

Exploration d’un espace d’états

Flexagone

But de l'activité : manipuler un flexagone pour déterminer exactement combien de faces il a.

Consigne

- Faire l'activité (si vous ne l'avez pas déjà faite)
- Situer la notion d'état dans les objectifs de l'activité
- Élaborer un discours pour institutionnaliser

Walgenwitz & Wack, Petit x n°119, 2023

Sudoku

Situation

Élaborer un solveur de Sudoku.

Consigne

- Se mettre d'accord sur les grandes lignes de l'élaboration du programme.
- Expliciter les états impliqués
  - état de la grille
  - état de l'exploration

Institutionnalisation

Glossaire

Posons ensemble les définitions de mots pertinents dans le contexte.

Une définition de l'état

- Analogie avec la physique : des éléments de description d'un système à un instant donné, avec le niveau d'abstraction adapté au contexte.
- Délimitation de ce qui constitue le système par opposition à l'environnement
- Nécessité d'un modèle opérationnel : la machine notionnelle
- Lien avec la notion de variable.