

Pattern matching

Tristan Crolard

Laboratoire CEDRIC
Equipe « Systèmes Sûrs »

`tristan.crolard@cnam.fr`

`cedric.cnam.fr/sys/crolard`

Pattern matching on lists – `is_empty`

Case analysis

- ▶ A list is either empty or not.

Example

$$\text{is_empty}(l) = \begin{cases} \text{True} & \text{if } l = [] \\ \text{False} & \text{otherwise} \end{cases}$$

```
>>> def is_empty[A](l: list[A]) -> bool:
      match (l):
        case []:
          return True
        case _:
          return False
```

```
>>> is_empty([])
```

True

```
>>> is_empty([1, 2, 3])
```

False

Pattern matching on lists – hd & tl

Case analysis

- ▶ A list is either empty or not.
- ▶ If it is not empty, it is composed of some **head** h (the first element) and some **tail** t (the remaining elements)

Example

$$\text{hd}(l) = \begin{cases} h & \text{if } l = [h, *t] \\ \text{undefined} & \text{otherwise} \end{cases}$$

```
>>> def hd[A](l: list[A]) -> A | None:
  match l:
    case [h, *t]:
      return h
    case _:
      return None
```

```
>>> hd([1, 2, 3])
```

1

```
>>> hd([])
```

Example

$$\text{tl}(l) = \begin{cases} t & \text{if } l = [h, *t] \\ \text{undefined} & \text{otherwise} \end{cases}$$

```
>>> def tl[A](l: list[A]) -> list[A] | None:
      match l:
        case [h, *t]:
          return t
        case _:
          return None
```

```
>>> tl([1, 2, 3])
```

```
[2, 3]
```

```
>>> tl([])
```

Alternative – hd & tl using exceptions

```
>>> def hd[A](l: list[A]) -> A:
      match l:
        case [h, *t]:
          return h
        case _:
          raise ValueError
```

```
>>> hd([1, 2, 3])
```

1

```
>>> hd([])
```

Traceback (most recent call last):

...

ValueError

```
>>> def t1[A](l: list[A]) -> list[A]:  
    match l:  
      case [h, *t]:  
        return t  
      case _:  
        raise ValueError
```

```
>>> t1([1, 2, 3])
```

```
[2, 3]
```

```
>>> t1([])
```

```
Traceback (most recent call last):
```

```
...
```

```
ValueError
```