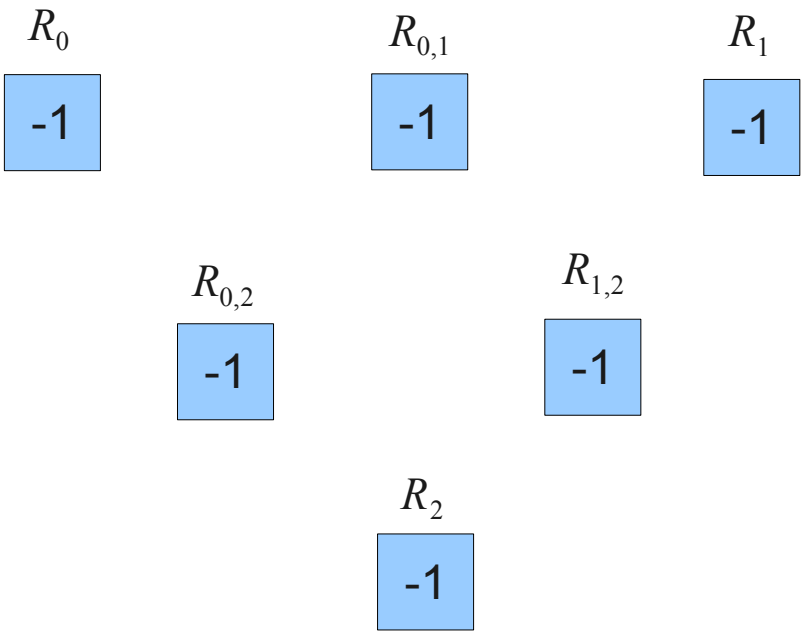
Distributed Systems

Exercise

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

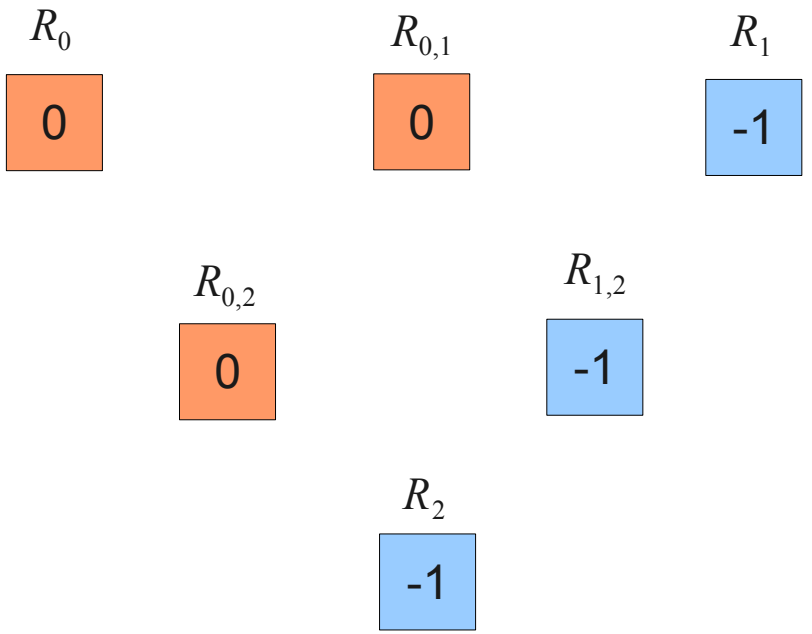
2. Writing multiple registers

- Start



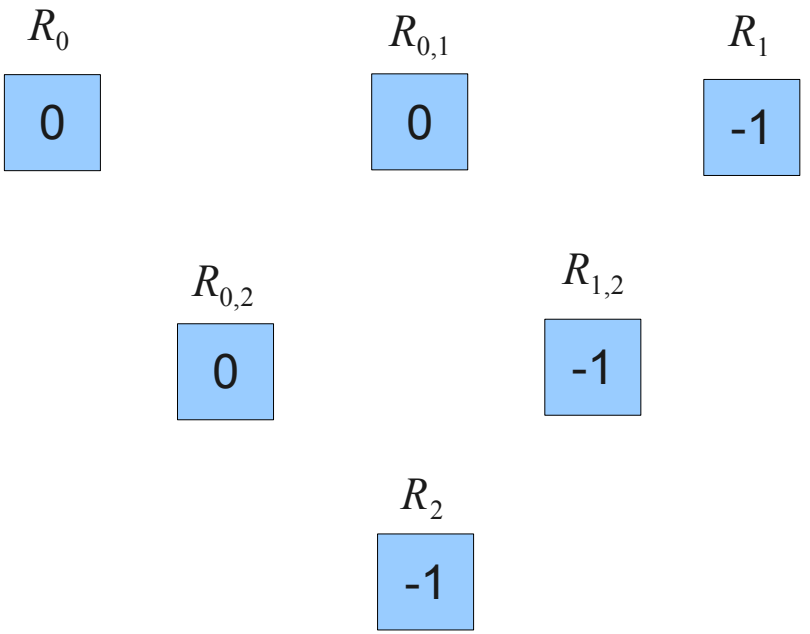
2. Writing multiple registers

- Write *id* into R_{id}, $R_{id,x}$, $R_{id,y}$



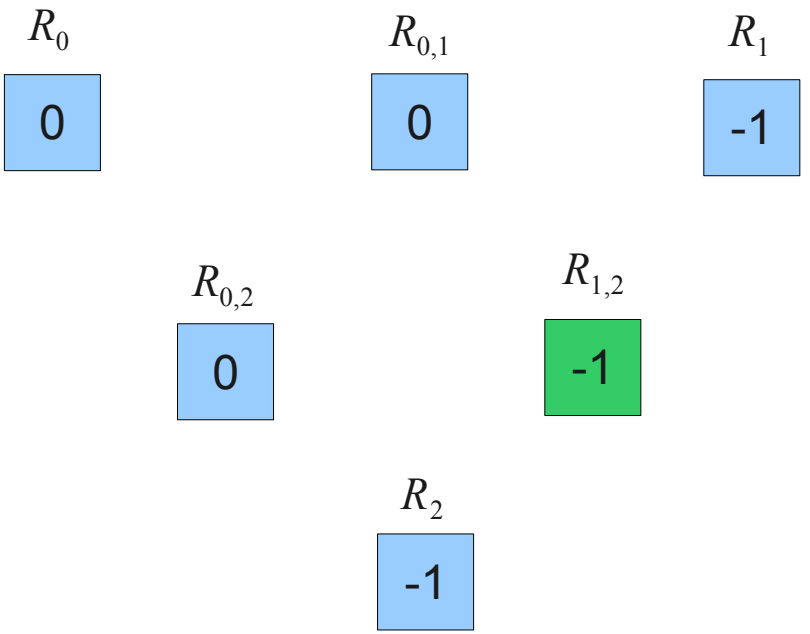
2. Writing multiple registers

- Write *id* into R_{id}, $R_{id,x}$, $R_{id,y}$
- For $i > j, i \in \{0,1,2\}, j \in \{1,2\}$
  - Find the faster one



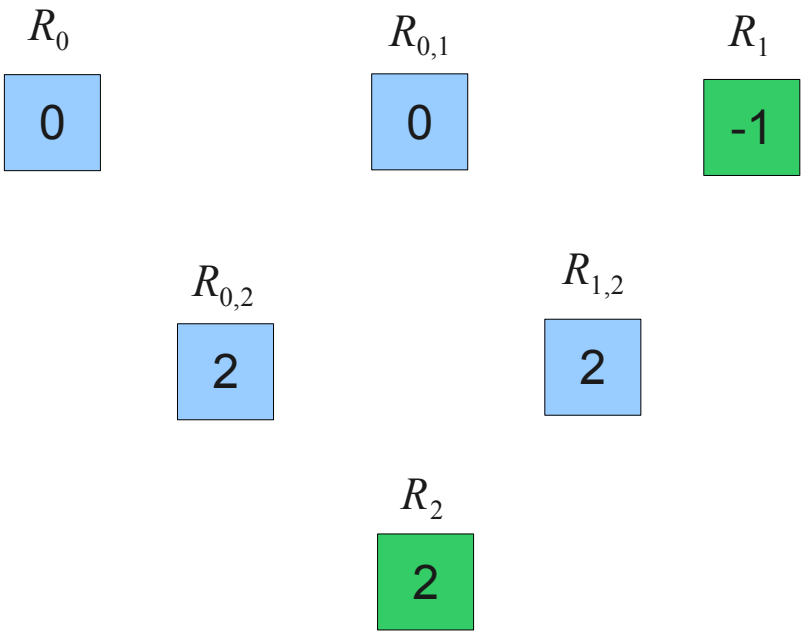
2. Writing multiple registers

- Write *id* into $R_{id}, R_{id,x}, R_{id,y}$
- For $i > j, i \in \{0,1,2\}, j \in \{1,2\}$
  - If $R_{i,j} = -1$ then neither is the fastest



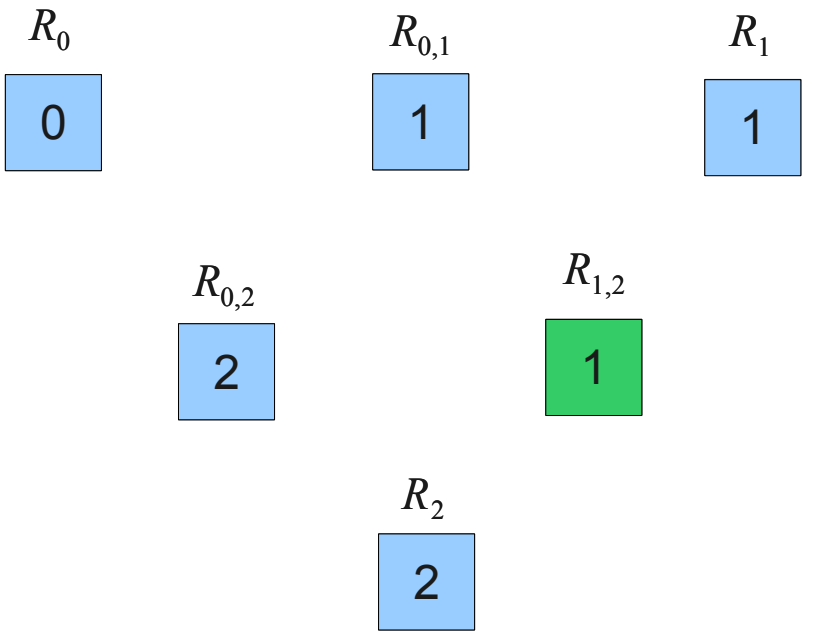
2. Writing multiple registers

- Write *id* into $R_{id}, R_{id,x}, R_{id,y}$
- For $i > j, i \in \{0,1,2\}, j \in \{1,2\}$
  - If $R_{i,j} = -1$ then neither is the fastest
  - If $R_i = -1$ then *j* is faster
  - If $R_j = -1$ then *i* is faster



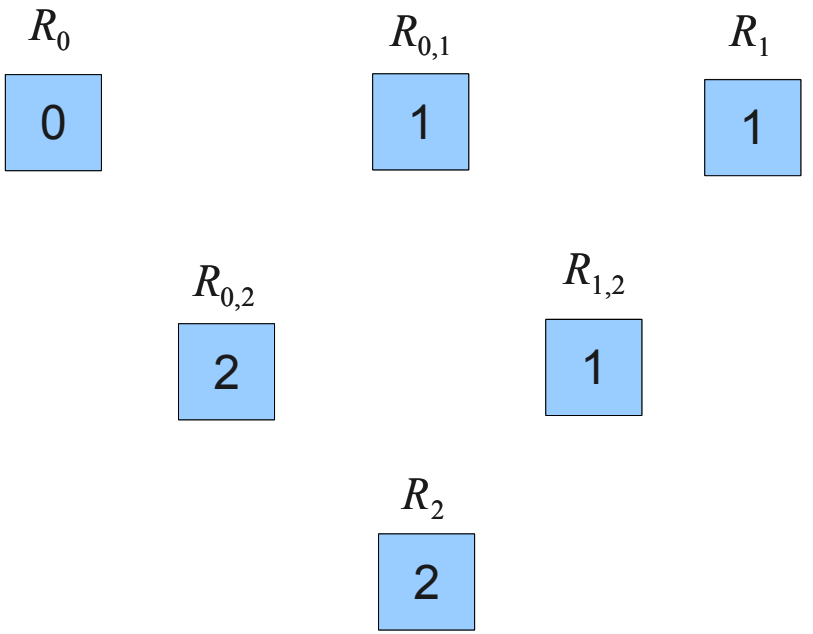
2. Writing multiple registers

- Write *id* into $R_{id}, R_{id,x}, R_{id,y}$
- For $i > j, i \in \{0,1,2\}, j \in \{1,2\}$
  - If $R_{i,j} = -1$ then neither is the fastest
  - If $R_i = -1$ then *j* is faster
  - If $R_j = -1$ then *i* is faster
  - If $R_{i,j} = i$ then *j* is faster
  - If $R_{i,j} = j$ then *i* is faster



2. Writing multiple registers

- Write *id* into $R_{id}, R_{id,x}, R_{id,y}$
- For $i > j, i \in \{0,1,2\}, j \in \{1,2\}$
  - If $R_{i,j} = -1$ then neither is the fastest
  - If $R_i = -1$ then *j* is faster
  - If $R_j = -1$ then *i* is faster
  - If $R_{i,j} = i$ then *j* is faster
  - If $R_{i,j} = j$ then *i* is faster
- Order the processes...



3. Analyze a protocol

```
s = '?', i = 0


decide(){
  id = this.getThreadId();

  value = s; // read s
  if( value == '?' ){
    s = input[ id ]; // write s
  }
  value = s; // read s

  if( value != input[ id ] ){
    decision = value;
  }
  else{
    if( i.fetchAndInc() == 1 )
      decision = input[ 1-id ];
    else
      decision = input[ id ];
  }
}
```

- Make it smaller!

3. Analyze a protocol

```
s = '?', i = 0


decide(){
  id = this.getThreadId();

  value = s; // read s
  if( value == '?' ){
    s = input[ id ]; // write s
  }
  value = s; // read s

  if( value != input[ id ] ){
    decision = value;
  }
  else{
    if( i.fetchAndInc() == 1 )
      decision = input[ 1-id ];
    else
      decision = input[ id ];
  }
}
```

- input[0] == input[1]
  - not interesting
- Let
  - input[0] = 0
  - input[1] = 1

3. Analyze a protocol

```
s = '?', i = 0
input[0] = 0, input[1] = 1

decide(){
  id = this.getThreadId();

  value = s; // read s
  if( value == '?' ){
    s = input[ id ]; // write s
  }
  value = s; // read s

  if( value != input[ id ] ){
    decision = value;
  }
  else{
    if( i.fetchAndInc() == 1 )
      decision = input[ 1-id ];
    else
      decision = input[ id ];
  }
}
```

- "Nice" case:
  - $value_0 = value_1$

3. Analyze a protocol

```
s = '?', i = 0
input[0] = 0, input[1] = 1

decide(){
  id = this.getThreadId();

  value = s; // read s
  if( value == '?' ){
    s = input[ id ]; // write s
  }
  value = s; // read s

  if( value != input[ id ] ){
    decision = value;
  }
  else{
    if( i.fetchAndInc() == 1 )
      decision = input[ 1-id ];
    else
      decision = input[ id ];
  }
}
```

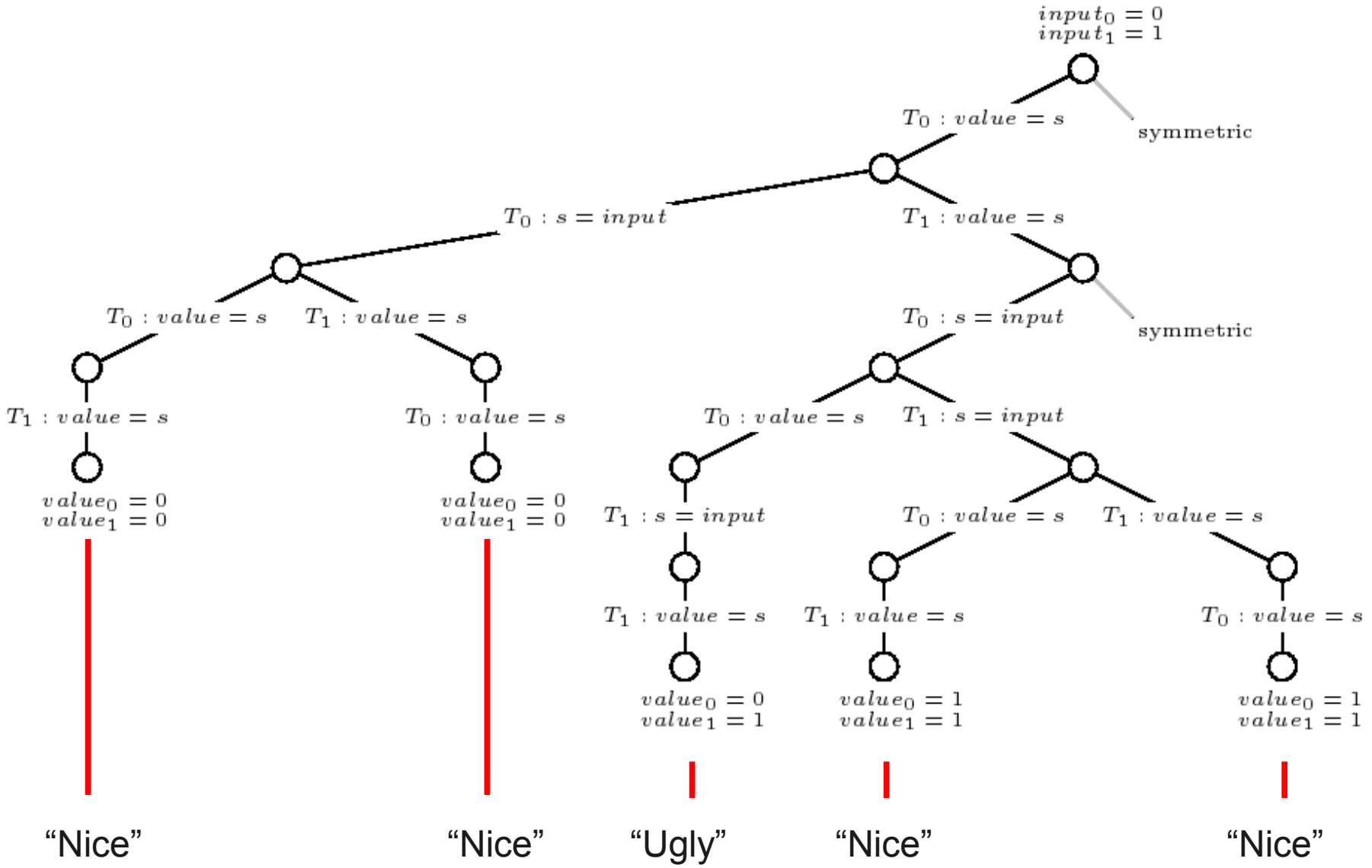
- "Nice" case:
  - $value_0 = value_1$
- "Ugly" case:
  - $value_0 = 0$
  - $value_1 = 1$

3. Analyze a protocol

```
s = '?', i = 0
input[0] = 0, input[1] = 1

decide(){
  id = this.getThreadId();

  value = s; // read s
  if( value == '?' ){
    s = input[ id ]; // write s
  }
  value = s; // read s

  if( value != input[ id ] ){
    decision = value;
  }
  else{
    if( i.fetchAndInc() == 1 )
      decision = input[ 1-id ];
    else
      decision = input[ id ];
  }
}
```

- "Nice" case:
  - $value_0 = value_1$
- "Ugly" case:
  - $value_0 = 0$
  - $value_1 = 1$
- "Error" case:
  - $value_0 = 1$
  - $value_1 = 0$

3. Analyze a protocol



Finish

- Questions
- Solve the exercise