

Mathematical programming with quantors

Victor Eijkhout, Susan Lindsey

Fall 2023

last formatted: February 6, 2024

1. Quantors / quantifier

Many math statements can be formulated with quantification:

- Universal quantification with 'for-all' quantor:

$$\forall_n: P(n)$$

- Existential quantification with 'there-is' quantor:

$$\exists_n: Q(n)$$

These two are equivalent in a way:

$$\neg \forall_n: S(n) \Leftrightarrow \exists_n: \neg S(n)$$

'if a statement is not true for all n , then there is a value n for which it is not true'. And the other way around.

2. For all: prime testing

The statement ' n is prime' can be formulated with universal quantification:

all f smaller than n don't divide in n

$$\text{isprime}(n) \equiv \forall_{2 \leq f < n} : \neg \text{divides}(f, n)$$

$$\text{isprime}(n) \equiv \neg \text{divides}(2, n) \cap \dots \cap \neg \text{divides}(n - 1, n)$$

3. Universal quantification in code

Translate the mathematical formulation to code:

```
for (int f=2; f<n; f++)  
    isprime = isprime && not divides(f,p)
```

Initialization: for-all over empty set is true:

```
bool isprime{true};  
for (int f=2; f<n; f++)  
    isprime = isprime && not divides(f,p)
```

4. Prime testing again

Per the equivalence note above:

n is prime if there is no divisor

$$\text{isprime}(n) \equiv \neg \exists_{2 \leq f < n} : \text{divides}(f, n)$$

To get a pure quantor, and not a negated one, we write

n is not a prime if there is a divisor

$$\text{isnotprime}(n) \equiv \exists_{2 \leq f < n} : \text{divides}(f, n)$$

Spelling out the exists-quantor as

$$\text{isnotprime}(n) \equiv \text{divides}(2, n) \cup \dots \cup \text{divides}(n-1, n)$$

5. In code

Translate the existential quantifier formulation into code taking care of the initial condition:

applying $\exists_{s \in S} P(s)$ over an empty set S is **false**:

```
bool isnotprime{false};  
for (int f=2; f<n; f++)  
    isnotprime = isnotprime or divides(f,p)  
bool isprime = not isnotprime;
```