Mathematical programming with quantors

Victor Eijkhout, Susan Lindsey

Fall 2023
last formatted: September 19, 2023

1. Quantors / quantifier

Many math statements can be formulated with quantification:

- Universal quantification with 'for-all' quantor:

$$\forall_n\colon P(n)$$

- Existential quantificaiton with 'there-is' quantor:

$$\exists_n\colon Q(n)$$

These two are equivalent in a way:

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

'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

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

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

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

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

To get a pure quantor, and not a negated one, we write
n is not a prime if there is a divisor

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

Spelling out the exists-quantor as

$$\mathrm{isnotprime}(n) \equiv \mathrm{divides}(2, n) \cup \ldots \cup \mathrm{divides}(n-1, n)$$

5. In code

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;
```