Many of the negative results that we have presented in the previous lecture immediately apply to Peano Arithmetic, which, as we have shown, can represent the computable functions over natural numbers. One may argue that this is the case because Peano Arithmetic has infinitely many (induction) axioms and that a finite axiom system surely wouldn't lead to undecidability and undefinability issues. In the following we will show that even a finite axiom system can be sufficiently strong to allow for a representation of all computable functions and that all the negative results carry over theories that are much smaller than Peano Arithmetic.

Q: *How can we give a finite axiomatization for a sufficiently strong theory of arithmetic?*

24.1 The Theory $\mathcal{Q}$

If we simply drop the induction axioms from Peano Arithmetic, the resulting theory would be extremely weak, as we wouldn't even be able to prove that every number different from zero must be the successor of some other number. However, theoretical investigations have shown adding this simple law in place of the induction scheme leads to an astonishingly strong formal theory. For historical reasons, the resulting theory is called $\mathcal{Q}$. It is defined as as the theory over the language $\mathcal{L}$(=,+,∗,0,1) that satisfies the following axioms

Successor Axioms

```
non-surjective   (∀x) ∼(x+1 = 0)
injective        (∀x,y) (x+1=y+1 ⊃ x=y)
non-zero         (∀x) ∼(x=0) ⊃ (∃y)(y+1=x)
```

Addition Axioms

```
add-base         (∀x) (x+0 = x)
add-step         (∀x,y) (x+(y+1) = (x+y)+1)
```

Multiplication Axioms

```
mul-base         (∀x) (x*0 = 0)
mul-step         (∀x,y) (x*(y+1) = (x*y)+x)
```

The original formulation of the theory $\mathcal{Q}$ only mentions these 7 axioms but assumes that function symbols and equality are built into first-order logic. Since our formulation of first-order logic does not make this assumption we have to add the functionality axioms for + and ∗, and the equality axioms `ref`, `sym`, and `trans`. We also need a restricted substitution axiom that permits substitution on the level of atomic formulas. Since atomic formulas are built from predicate symbols, variables, parameters, and – in a theory with functions and equality – function applications and the equality predicate, we add a term-substitution axiom for every argument of every function symbol.

term-subst: $\quad$ (∀x,y) (x=y ⊃ f(.,x,.) = f(.,y,.))

Since the language only provides two function symbols (all others would be an abbreviation for combinations of these) there are only four substitution axioms.

term-subst$_{+l}$:	(∀x,y,z) (x=y ⊃ x+z = y+z
term-subst$_{+r}$:	(∀x,y,z) (x=y ⊃ z+x = z+y
term-subst$_{*l}$:	(∀x,y,z) (x=y ⊃ x*z = y*z
term-subst$_{*r}$:	(∀x,y,z) (x=y ⊃ z*x = z*y

This means that the theory $\mathcal{Q}$ is finitely axiomatizable.

This theory is obviously consistent with our view of the natural numbers, because the *Standard-Interpretation* works out fine. It has the advantage of having only finitely many axioms, and we will show that it is very strong, as far as expressiveness is concerned.

In fact, a careful look at the proof that all computable functions can be represented in Peano Arithmetic shows that it did not depend on the induction axiom at all. Recall that an n-ary function $f:\mathbb{N}\to\mathbb{N}$ is representable in $\mathcal{Q}$ if there is an (n+1)-ary predicate R_f in $\mathcal{L}(\texttt{=,+,*,0,1})$ such that for all $x_1,..,x_n,y\in\mathbb{N}$

- $f(x_1,..,x_n)=y$ implies $\models_T R_f(\underline{\boldsymbol{x_1}},\ldots,\underline{\boldsymbol{x_n}},\underline{\boldsymbol{y}})$
- $f(x_1,..,x_n)\neq y$ implies $\models_T \sim R_f(\underline{\boldsymbol{x_1}},\ldots,\underline{\boldsymbol{x_n}},\underline{\boldsymbol{y}})$

What matters is that we can find an appropriate predicate to represent a computable function and that we can show semantically that the above conditions hold in every model of $\mathcal{Q}$. We are free to prove this by (semantic) induction on numbers and are not forced to use the induction *axiom*. Let us briefly review how we represent successor, constants, projection, composition, minimization, and primitive recursion.

- The *successor function* can be represented by the formula $x\texttt{+1=}y$.
- *Addition* $+$ is represented by the formula $x\texttt{+}y\texttt{=}z$.
- *Multiplication* $*$ is represented by the formula $x\texttt{*}y\texttt{=}z$.
- The *constant function* c_k can be represented by the formula $y\texttt{=}\underline{\boldsymbol{k}}$.
- The *projection function* $\pi^n{}_i$ can be represented by the formula $y\texttt{=}x_i$.
- The *composition* $h = g\circ f_1,..,f_k$ can be represented by the formula
  $(\exists z_1,..z_k)(\ \mathtt{R}_{f_1}(x_1,..x_n,z_1)\ \wedge\ldots\wedge\ \mathtt{R}_{f_k}(x_1,..x_n,z_k)\ \wedge\ \mathtt{R}_g(z_1,..z_k,y))$,
  where $\mathtt{R}_{f_1},\ldots,\mathtt{R}_{f_k}$, and $\mathtt{R}_g$ are the predicates representing $f_1,..,f_k$, and g respectively.
- The *minimization* $h = \mu f$ can be represented by the formula below, where $\mathtt{R}_f$ represents f:
  $(\forall \mathtt{z})(\mathtt{z}\leq y\ \supset\ (\mathtt{R}_f(x_1,..x_n,\mathtt{z},0)\ \Leftrightarrow\ \mathtt{z}=y))$.
- *Primitive recursion* can be expressed entirely in all the functions and constructors mentioned above. The proof for that was lengthy but depended only on the properties of these functions and not on the specific logic used for representing μ-recursive functions.

Thus we can use exactly the same formulas that we used in the representation of computable functions in Peano Arithmetic to represent the computable functions in $\mathcal{Q}$. Checking the validity of these representations does not depend on the induction axiom (we actually just need an instance of `non-zero`), since induction is only required on the meta-level but not on the object level of our theory. Thus we get

Theorem: All μ-recursive functions are representable in $\mathcal{Q}$.

This means that all the negative results about theories that can represent computable functions apply to the theory $\mathcal{Q}$ as well.

Theorem: $\mathcal{Q}$ is undecidable and so is every consistent extension of $\mathcal{Q}$.

Theorem: No consistent, axiomatizable extension of $\mathcal{Q}$ is complete.

We are now even able to prove that first-order logic is undecidable. We prove this by reducing decidability in first-order logic to decidability in $\mathcal{Q}$.

Theorem: [Church's Theorem] Validity in first-order logic is undecidable.

Proof: Let Ax_Q be the conjunctions of all the axioms of the theory $\mathcal{Q}$. Then a formula X is a theorem in $\mathcal{Q}$ if and only if the formula $Ax_Q \supset X$ is valid (in first-order logic).

Now assume that there is a procedure *decide-fol* for deciding validity in first-order logic. Then the algorithm that translates a formula X into $Ax_Q \supset X$ and then applies *decide-fol* to this formula would decide validity in $\mathcal{Q}$. Since $\mathcal{Q}$ is undecidable, this cannot be. □

An interesting consequence of Church's Theorem is that *first-order logic is incomplete (as a theory)*, because it is obviously consistent and axiomatizable but not decidable. This, however, is not surprising. Since there is an unlimited number of models for first-order logic, there are plenty of first-order formulas that are not valid and whose negation isn't valid either.

24.2 Models of $\mathcal{Q}$

Although the theory $\mathcal{Q}$ is expressive enough to represent all computable functions and thus appears to be as strong as Peano Arithmetic, the fact that we removed the induction axiom will obviously have some effect on what is provable in $\mathcal{Q}$. Here is one example.

The formula `(∀x)(x+1≠x)` is not valid in $\mathcal{Q}$.

At a first glance this claim may appear strange, since $x+1 \neq x$ is one of the basic laws of the natural numbers and the formula can easily be proven in Peano Arithmetic. However, one has to keep in mind that there are many non-standard models for $\mathcal{Q}$ that violate some of the basic laws of the natural numbers. Many of these models are not models of Peano Arithmetic, since the induction axiom essentially states that any property of what we can show for natural numbers must hold for all numbers, standard or not. In the theory $\mathcal{Q}$, we don't have this axiom anymore.

Q: *How can we prove the above claim?*

We have to construct a model of $\mathcal{Q}$ in which the law $x+1 \neq x$ does not hold. Since it is true for natural numbers, we have to construct a non-standard model that has extra elements besides natural numbers and satisfies the axioms of $\mathcal{Q}$ on these extra elements, but not the formula `(∀x)(x+1≠x)`.

So let us pick an element ω that is not a natural number. We don't need to give an intuition for this element, but we could imagine that ω could be infinity. There is *no term that represents* ω, but nevertheless ω must be considered when we evaluate a universally quantified formula.

On the natural numbers, all symbols of $\mathcal{Q}$ will be interpreted as usual. `0` remains to be interpreted as zero, `1` as one, `+` as addition, and `*` as multiplication. But we define $i+\omega = \omega = \omega+i$ for all i, $0*\omega = 0 = \omega*0$, and $i*\omega = \omega = \omega*i$ for all other i. The following table summarizes the interpretation of the symbols `+` and `*`.

$+$	$j \in \mathbb{N}$	ω	$*$	0	$j \neq 0$	ω
$i \in \mathbb{N}$	$i+j$	ω	0	0	0	0
ω	ω	ω	$i \neq 0$	0	$i*j$	ω
			ω	0	ω	ω

Note that this table is a semantical table about *one specific interpretation* of the symbols `+` and `*`, not about the symbols themselves.

It is obvious that the law $x+1 \neq x$ does not hold in this interpretation, as $\omega+1 = \omega$. What needs to be done is verifying that the interpretation is in fact a model of $\mathcal{Q}$. Since we are using the standard interpretation of `+` and `*` as long as we are dealing with natural numbers, we only have to check the seven axioms when ω is involved.

`non-surjective`: $\omega+1 \neq 0$ is obvious from the above multiplication table.

`injective`: $x+1 = \omega+1$ implies $x+1 = \omega$ which is the case only if $x = \omega$

Since the addition table is commutative, $\omega+1 = x+1$ implies $\omega = x$.

`non-zero`: Since $\omega+1 = \omega$ we know that $(\exists y)(y+1 = \omega)$

`add-base`: $\omega+0 = \omega$ by definition.

`add-step`: $\omega+(y+1) = \omega = \omega+1 = (\omega+y)+1$

$x+(\omega+1) = x+\omega = \omega = \omega+1 = (x+\omega)+1$

`mul-base`: $\omega*0 = 0$ by definition

`mul-step`: $\omega*(y+1) = \omega = (\omega*y)+\omega$ for any y

$0*(\omega+1) = 0 = 0+0 = (0*\omega)+0$

$x*(\omega+1) = x*\omega = \omega = \omega+x = (x*\omega)+x$ for all other x

Thus the given interpretation is a model of $\mathcal{Q}$ that violates the law $x+1 \neq x$, which means that `(∀x)(x+1≠x)` is not valid in $\mathcal{Q}$.