# Basics of Induction (Review)

# Induction

Principle of *ordinary induction* on natural numbers:

> *Suppose that* $P$ *is a predicate on the natural numbers. Then:*
>
> *If* $P(0)$
> *and, for all* $i$, $P(i)$ *implies* $P(i+1)$,
> *then* $P(n)$ *holds for all* $n$.

# Example

Theorem: $2^0 + 2^1 + ... + 2^n = 2^{n+1} - 1$, for every $n$.
Proof: Let $P(i)$ be "$2^0 + 2^1 + ... + 2^i = 2^{i+1} - 1$."

- Show $P(0)$:

$$2^0 = 1 = 2^1 - 1$$

- Show that $P(i)$ implies $P(i+1)$:

$$\begin{aligned} 2^0 + 2^1 + ... + 2^{i+1} &= (2^0 + 2^1 + ... + 2^i) + 2^{i+1} \\ &= (2^{i+1} - 1) + 2^{i+1} && \text{by IH} \\ &= 2 \cdot (2^{i+1}) - 1 \\ &= 2^{i+2} - 1 \end{aligned}$$

- The result ($P(n)$ for all $n$) follows by the principle of (ordinary) induction.

## Shorthand form

Theorem: $2^0 + 2^1 + ... + 2^n = 2^{n+1} - 1$, for every $n$.
Proof: By induction on $n$.

- Base case ($n = 0$):

$$2^0 = 1 = 2^1 - 1$$

- Inductive case ($n = i + 1$):

$$\begin{array}{rcll} 2^0 + 2^1 + ... + 2^{i+1} & = & (2^0 + 2^1 + ... + 2^i) + 2^{i+1} & \\ & = & (2^{i+1} - 1) + 2^{i+1} & \text{IH} \\ & = & 2 \cdot (2^{i+1}) - 1 & \\ & = & 2^{i+2} - 1 & \end{array}$$

# Complete Induction

Principle of *complete induction* on natural numbers:

> *Suppose that $P$ is a predicate on the natural numbers. Then:*
>
> *If, for each natural number $n$,*
> *given $P(i)$ for all $i < n$*
> *we can show $P(n)$,*
> *then $P(n)$ holds for all $n$.*

# Complete versus ordinary induction

Ordinary and complete induction are *interderivable* — assuming one, we can prove the other.

Thus, the choice of which to use for a particular proof is purely a question of style.

We'll see some other (equivalent) styles as we go along.

# Syntax

# Simple Arithmetic Expressions

Here is a BNF grammar for a very simple language of arithmetic expressions:

| | | |
|---|---|---|
| `t` ::= | | *terms* |
| | `true` | *constant true* |
| | `false` | *constant false* |
| | `if t then t else t` | *conditional* |
| | `0` | *constant zero* |
| | `succ t` | *successor* |
| | `pred t` | *predecessor* |
| | `iszero t` | *zero test* |

Terminology:

- `t` here is a *metavariable*

# Abstract vs. concrete syntax

Q: Does this grammar define a set of *character strings*, a set of *token lists*, or a set of *abstract syntax trees*?

# Abstract vs. concrete syntax

Q: Does this grammar define a set of *character strings*, a set of *token lists*, or a set of *abstract syntax trees*?

A: In a sense, all three. But we are primarily interested, here, in abstract syntax trees.

For this reason, grammars like the one on the previous slide are sometimes called *abstract grammars*. An abstract grammar *defines* a set of abstract syntax trees and *suggests* a mapping from character strings to trees.

We then *write* terms as linear character strings rather than trees simply for convenience. If there is any potential confusion about what tree is intended, we use parentheses to disambiguate.

Q: So, are

```
succ 0
succ (0)
(((succ (((((0))))))))
```

"the same term"?

What about

```
succ 0
pred (succ (succ 0))
```

?

# A more explicit form of the definition

The set $\mathcal{T}$ of *terms* is the smallest set such that

1. $\{\texttt{true}, \texttt{false}, 0\} \subseteq \mathcal{T}$;
2. if $\texttt{t}_1 \in \mathcal{T}$, then $\{\texttt{succ t}_1, \texttt{pred t}_1, \texttt{iszero t}_1\} \subseteq \mathcal{T}$;
3. if $\texttt{t}_1 \in \mathcal{T}$, $\texttt{t}_2 \in \mathcal{T}$, and $\texttt{t}_3 \in \mathcal{T}$, then $\texttt{if t}_1 \texttt{ then t}_2 \texttt{ else t}_3 \in \mathcal{T}$.

# Inference rules

An alternate notation for the same definition:

$$\texttt{true} \in \mathcal{T} \qquad \texttt{false} \in \mathcal{T} \qquad \texttt{0} \in \mathcal{T}$$

$$\frac{\texttt{t}_1 \in \mathcal{T}}{\texttt{succ t}_1 \in \mathcal{T}} \qquad \frac{\texttt{t}_1 \in \mathcal{T}}{\texttt{pred t}_1 \in \mathcal{T}} \qquad \frac{\texttt{t}_1 \in \mathcal{T}}{\texttt{iszero t}_1 \in \mathcal{T}}$$

$$\frac{\texttt{t}_1 \in \mathcal{T} \qquad \texttt{t}_2 \in \mathcal{T} \qquad \texttt{t}_3 \in \mathcal{T}}{\texttt{if t}_1 \texttt{ then t}_2 \texttt{ else t}_3 \in \mathcal{T}}$$

Note that "the smallest set closed under..." is implied (but often not stated explicitly).

Terminology:

- axiom vs. rule
- concrete rule vs. rule scheme