csci54 – discrete math & functional programming
proofs continued

on proof writing

- proof: a convincing argument written for a particular audience

- guidelines:
  - unless it's a direct proof without cases, state what proof technique you're using
  - define variables
  - have a concluding statement

proving "for all" statements

- claim: if x and y are even integers, then x+y is an even integer
- claim: given any two integers x and y, if x and y are even then x+y is even.

- observation on proving "for all" statements
  - "let x be an element of S"
  - since true for any element of S, must be true for all elements of S

Above all, remember that your primary goal in writing is communication. Just as when you are programming, it is possible to write two solutions to a problem that both “work,” but which differ tremendously in readability. Document! Comment your code; explain why this statement follows from previous statements. Make your proofs—and your code!—a pleasure to read.

CDMCS – end of section 4.3

direct proof : example (v1)

- claim: If a number is odd, then its binary representation ends with a 1.

- proof:
  - Let k be an arbitrary odd integer.
  - Then there exists an integer r such that $k=2r+1$.
  - Now let $d_n \ldots d_2 d_1 d_0$ be the binary representation of r.
  - The binary representation of 2r is then $d_n \ldots d_2 d_1 d_0 0$, and
  - The binary representation of $k=2r+1= d_n \ldots d_2 d_1 d_0 1$.

- conclusion: Therefore the binary representation of any odd integer ends with a 1.

direct proof : example (v2)

- claim: If a number is odd, then its binary representation ends with a 1.
- proof:
  - Let k be an arbitrary odd integer.
  - Then there exists an integer r such that k=2r+1.
  - Now let $d_n \ldots d_2 d_1 d_0$ be the binary representation of r.
    - This means r = ...
    - So 2r = ...
  - The binary representation of 2r is therefore $d_n \ldots d_2 d_1 d_0 0$, and
  - The binary representation of $k=2r+1= d_n \ldots d_2 d_1 d_0 1$.
- conclusion: Therefore the binary representation of any odd integer ends with a 1.

direct proof : example (v3)

- claim: If a number is odd, then its binary representation ends with a 1.
- proof:
  - Let k be an arbitrary odd integer.
  - Then there exists an integer r such that k=2r+1.
  - Now let $d_n \ldots d_2 d_1 d_0$ be the binary representation of r.
    - This means r = ...
    - So 2r = ...

      = ...
  - The binary representation of 2r is therefore $d_n \ldots d_2 d_1 d_0 0$, and
  - The binary representation of k=2r+1= $d_n \ldots d_2 d_1 d_0 1$.
- conclusion: Therefore the binary representation of any odd integer ends with a 1.

if and only if: example

- prove the following claim by proving each direction separately. Use a direct proof in one direction and a proof of the contrapositive in the other.

- claim: For all integers j and k, j and k are odd if and only if their product jk is odd.
- proof: Let j and k be arbitrary integers.
  - () If j and k are odd, then jk is odd
  - () If jk is odd, then j and k are odd
  - Therefore for all integers j and k, j and k are odd if and only if jk is odd.

a way that things can go wrong

- Claim: 1=0

Proof. Suppose that $1 = 0$. Then:

$$
\begin{aligned}
& & 1 &= 0 \\
&\text{therefore, multiplying both sides by } 0 & 0 \cdot 1 &= 0 \cdot 0 \\
&\text{and therefore,} & 0 &= 0. \quad \checkmark
\end{aligned}
$$

And, indeed, $0 = 0$. Thus the assumption that $1 = 0$ was correct, and the theorem follows. □

- More examples, discussion in Chapter 4.5 of the book

- Material about proofs of plus a 0 = plus 0 a, plus a 1 = plus 1 a, plus a b = plus b a for haskell:
- plus 0 b = b
- plus a b = (plus (a-1) b) + 1

proof techniques

- direct proof:
  - start with known facts. repeatedly infer additional new facts until can conclude what you want to show.
  - may divide work into cases
- proof of the contrapositive:
  - if trying to prove an implication, prove the contrapositive instead
- proof by contradiction
  - Claim: p is logically equivalent to $\neg p \rightarrow \bot$

proof techniques

- direct proof:
  - start with known facts. repeatedly infer additional new facts until can conclude what you want to show.
  - may divide work into cases
- proof of the contrapositive:
  - if trying to prove an implication, prove the contrapositive instead
- proof by contradiction
  - if trying to prove a statement, assume the statement is not true and prove something that is clearly false. From this conclude that the original statement must be true.

proof by contradiction – logic and example

- the proposition p is logically equivalent to $\neg p \rightarrow \bot$
- claim: The statement $\exists y: \forall x: y > x$ is false.
- proof by contradiction:
  - assume the statement is True; we'll show this leads to a contradiction
  - let y* be a y for which the statement is True.
  - then y* must be larger than all real numbers x.
  - however, y* is also a real number, so y* > y*.
  - this is a contradiction so the assumption that the statement is True must be wrong.
  - therefore the original statement is False.

Example from csci101

Theorem: If L is a context-free language, then:

$$\exists k \geq 1\ (\forall \text{ strings } w \in L, \text{ where } |w| \geq k\ (\exists u, v, x, y, z\ \ (w = uvxyz,\ vy \neq \varepsilon,\ |vxy| \leq k, \text{ and } \forall q \geq 0\ (uv^q xy^q z \text{ is in } L)))).$$

- used to prove that a language L is **not** context free
  - proof by contradiction: assume that L is context free. Then there must be a value k that satisfies the above theorem.
  - now show that such a k cannot exist