

## Lecture 1: Jan 2, 2014

*Instructor: Parag Singla**Scribes: scribe-name*

**Note:** *LaTeX* template courtesy of UC Berkeley EECS dept.

**Disclaimer:** *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

This lecture's notes illustrate some uses of various  $\text{\LaTeX}$  macros. Take a look at this and imitate.

## 1.1 Some theorems and stuff

We now delve right into the proof.

**Lemma 1.1** *This is the first lemma of the lecture.*

**Proof:** The proof is by induction on  $\dots$ . For fun, we throw in a figure.

Figure 1.1: A Fun Figure

This is the end of the proof, which is marked with a little box. ■

### 1.1.1 A few items of note

Here is an itemized list:

- this is the first item;
- this is the second item.

Here is an enumerated list:

1. this is the first item;
2. this is the second item.

Here is an exercise:

**Exercise:** Show that  $P \neq NP$ .

Here is how to define things in the proper mathematical style. Let  $f_k$  be the *AND* – *OR* function, defined by

$$f_k(x_1, x_2, \dots, x_{2^k}) = \begin{cases} x_1 & \text{if } k = 0; \\ \text{AND}(f_{k-1}(x_1, \dots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \dots, x_{2^k})) & \text{if } k \text{ is even;} \\ \text{OR}(f_{k-1}(x_1, \dots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \dots, x_{2^k})) & \text{otherwise.} \end{cases}$$

**Theorem 1.2** *This is the first theorem.*

**Proof:** This is the proof of the first theorem. We show how to write pseudo-code now.

Consider a comparison between  $x$  and  $y$ :

```

if  $x$  or  $y$  or both are in  $S$  then
    answer accordingly
else
    Make the element with the larger score (say  $x$ ) win the comparison
    if  $F(x) + F(y) < \frac{n}{t-1}$  then
         $F(x) \leftarrow F(x) + F(y)$ 
         $F(y) \leftarrow 0$ 
    else
         $S \leftarrow S \cup \{x\}$ 
         $r \leftarrow r + 1$ 
    endif
endif

```

This concludes the proof. ■

## 1.2 Next topic

Here is a citation, just for fun [CW87].

## References

- [CW87] D. COPPERSMITH and S. WINOGRAD, “Matrix multiplication via arithmetic progressions,” *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 1–6.