Research Topics in SE

Zhendong Su
Objectives

- Learn to present technical work
- Learn to understand & evaluate research papers
- Learn several key research directions in the area
Objectives

- Learn to present technical work
- Learn to understand & evaluate research papers
- Learn several key research directions in the area
- Have a good deal of fun doing so
Paper Pool for Spring 2020

2. Executable formal semantics for the POSIX shell. POPL 2020.
23. Rethinking Debugging as Productive Failure for CS Education. SIGCSE 2019.
32. HOPPITY: LEARNING GRAPH TRANSFORMATIONS TO DETECT AND FIX BUGS IN PROGRAMS. ICLR 2020.
Preparing a Talk

- Check your presentation date
- Study your paper(s)
- Create draft presentation
- Meet advisor, get feedback
Preparing a Talk: **Start Early**

- Preparing a good presentation takes time
- So, please start early!

1. Check your presentation date
2. Study your paper(s)
3. Create draft presentation
4. Meet advisor, get feedback

---

Research Topics in SE
Preparring a Talk: **Study Paper**

- **3 C’s of reading**
  - *Carefully*: look up terms, possibly read cited papers
  - *Critically*: find limitations, flaws
  - *Creatively*: think of improvements

- **Try examples** by hand
- **Try tools** if available
- **Consult advisors for questions**
Preparing a Talk: **Create Draft**

- Explain the work’s **motivation**
- Clearly present the technical solution and results
  - Use **own examples**, not the ones in the paper
  - Include a demo if appropriate
- Discuss **limitations** or improvements
- Focus on **key concepts**
  - Do not present all the details

---

1. **Check your presentation date**
2. **Study your paper(s)**
3. **Create draft presentation**
4. **Meet advisor, get feedback**
Preparing a Talk: **Get Feedback**

- **Prepare** for the meeting
  - Schedule early
  - Send slides in advance
  - Write down questions
- **Do** address feedback
  - Take notes
- **Meeting is mandatory!**
  - At least 1 week before the talk
Grading

- Quality of your presentation
  - How well did you understand the material?
  - How well did you present it?
  - How well did you answer the questions?

- Participation
  - Did you ask good questions?
  - Did you attend all sessions?

- We will take into account
  - the difficulty of the paper
  - suggestions you received from your TA advisor
  - time you had to prepare
Feedback

- Discuss your talk’s strengths/weaknesses in-class
  - Let us know upfront if you prefer us not to

- Arrange meeting with your advisor to get feedback
Schedule

- Meet once per week with ~2 presentations each time
  - Next meeting on March 10th

- Detailed schedule will be published online shortly
  - Including names of advisors/mentors
Your Talk: **Timing**

- 30 min for talk
  - 1.5 ~ 2 min per slide
- 15 min for Q&A and discussion

- The pace of talk is important
  - Too fast, the audience cannot follow
  - Too slow, people can get bored

- Practice your talk
  - Checkpoint after ~10 minutes
Your Talk: **Examples**

- Examples are crucial for understanding
  - Both yours and the audience’s
  - Prepare your own examples

- Try to find a running example
  - For motivation, problem, solution
  - Explain in detail (takes time)

- Reduce code example to the absolute necessary
  - Most people hate reading code
  - Use visualizations
Your Talk: **Design**

**Ilvm bug autopsy**

```c
struct tiny { char c; char d; char e; }

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

- **GVN:** load struct using 32-bit load
- **SRoA:** read past the struct’s end → undefined behavior

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```
Powerpoint vs. Latex

- **Powerpoint**
  - Visualizations and animations are easy
  - Don’t over-do it!

- **Latex**
  - Visualizations and animations are painful
  - Don’t under-do it!
Your Talk: **Avoid Frequent Mistakes**

- Don’t try to present all details
  - Focus on the key messages: motivation, problem, main idea, main result

- Don’t stare at the screen or your laptop
  - Look at the audience

- Come prepared
  - Study the paper in depth
  - Rehearse your talk
References

Markus Püschel’s small guide on giving talks


Highly recommended!
Paper Pool for Spring 2020

2. Executable formal semantics for the POSIX shell. POPL 2020.
23. Rethinking Debugging as Productive Failure for CS Education. SIGCSE 2019.
32. HOPPITY: LEARNING GRAPH TRANSFORMATIONS TO DETECT AND FIX BUGS IN PROGRAMS. ICLR 2020.