

16.7

An Application of directed graphs to make

Make/Makefile

- Ⓐ I know what make/makefile is.
- Ⓑ I do NOT know what make/makefile is.

make Utility [Feldman]

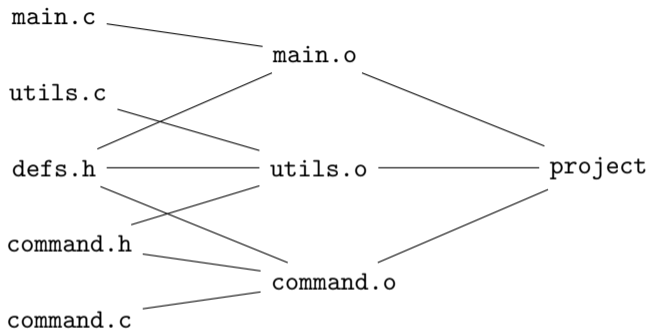
- ① Unix utility for automatically building large software applications
- ② A makefile specifies
 - ① Object files to be created,
 - ② Source/object files to be used in creation, and
 - ③ How to create them

An Example makefile

```
project: main.o utils.o command.o
    cc -o project main.o utils.o command.o

main.o: main.c defs.h
    cc -c main.c
utils.o: utils.c defs.h command.h
    cc -c utils.c
command.o: command.c defs.h command.h
    cc -c command.c
```

makefile as a Digraph



Computational Problems for `make`

- 1 Is the `makefile` reasonable?
- 2 If it is reasonable, in what order should the object files be created?
- 3 If it is not reasonable, provide helpful debugging information.
- 4 If some file is modified, find the fewest compilations needed to make application consistent.

Algorithms for make

- 1 Is the makefile reasonable? Is G a DAG?
- 2 If it is reasonable, in what order should the object files be created? Find a topological sort of a DAG.
- 3 If it is not reasonable, provide helpful debugging information. Output a cycle. More generally, output all strong connected components.
- 4 If some file is modified, find the fewest compilations needed to make application consistent.
 - 1 Find all vertices reachable (using DFS/BFS) from modified files in directed graph, and recompile them in proper order. Verify that one can find the files to recompile and the ordering in linear time.

THE END

...

(for now)