

CS6600 Homework 4

Matthew Whitesides

I. CHAPTER 3 PROBLEMS

- 7) Lemma 3.1 shows how through the use of a newly created object, two subjects with one having to take rights over the other can work together to take a right one of the subjects has over a third subject/object. If **X** were an object, however, the first step would not work as **X** could not create the new vertex **V**. This would not allow **X** ever to have a connection with an object it has *tg* privileges over, so **Z** would not be able to take the grant privileges from that object and share alpha right with it.
- 9) Because $s' = s$ or s' and $x_n = s'$ and x_i are all connected by label $t, g, bridge$ any of the three options for subject x will be able to take or pass any right from x_n . Then since there is a sequence of subjects where eventually $x_n = s'$ and s' has *tg* over s which in turn has α over y , x can obtain α from y .
 - If there is an edge from **x** to **y** labeled α can share is true.
 - Otherwise **s'** acquires α rights over **y** from **s** using the definition of terminal span.
 - **x'** acquires α rights over **y** from **s'** using the constraint that all the islands have *t,g*, or bridges between them.
 - **x'** gives α rights over **y** to **x** using the definition of initial span.
- 10) If we reverse the edge direction from **d** to **e** then *can share* is still true using the following witness.
 - **c** takes (*g* to **e**) from **d**.
 - **c** grants (*g* to **b**) to **e**.
 - **e** grants (*r* to **z**) to **b**.
 - **b** grants (*r* to **z**) to **a**.
 - **x** takes (*r* to **z**) from **a**.
- 11) If you dropped *take* from the TG model you could achieve a similar result from reversing the direction of all *t* edges and giving them grant (as problem 10 shows), however I feel this would less accurately represent the real world modeling of the system and the attack that is happening. At the same time I imagine similar to what SPM is trying to achieve would simplify the proofs a bit.
- 12) The acyclic creates constraints on the types of subjects created but not objects because inherently, the cycle is worried about child subjects creating other subjects/objects with rights inherited from their parents making a cycle. In contrast, objects created cannot create further objects/subjects, so the cycle ends there for that branch.

ACKNOWLEDGMENT

The author would like to thank Professor Bruce McMillin with the Department of Computer Science, Missouri University of Science and Technology.