Lecture Session 11: Janury 26, 2021

Notation

macros for copy-and-paste:

- $\texttt{purge}_u(\alpha)$
- $\texttt{obs}_u(s)$
- $\texttt{step}(s, a)$

Transitive Policies

- $L_1 \to L_2$
- $L_2 \to L_3$
- transitivity would mean: this "automatically" gives us $L_1 \to L_3$

intransitive policies: "L_2 may talk to L_3, but not about the stuff he learned from L_1".

purge function

- $D = \{L_1, L_2, H_1, H_2\}$
- policy: $L_1 \to L_2$, $L_2 \to L_1$, $H_1 \to H_2$, $H_2 \to H_1$, $L_1 \to H_1$, $L_1 \to H_2$, $L_2 \to H_1$, $L_2 \to H_2$
- $A = \{l_1, l_2, h_1, h_2\}$
- $\texttt{dom}(l_x) = L_x$, $\texttt{dom}(h_x) = H_x$
- $\alpha = l_1 l_2 h_1 h_1 l_1 l_2 h_1$
- $\texttt{purge}_{L_1}(\alpha) =?$
- idea: $\texttt{purge}_{L_1}(\alpha)$ contains exactly those actions from α that
  - "L_1 is allowed to see."
  - i.e., actions a that are performed by some agent v with $v \to L_1$
  - i.e., actions a with $\texttt{dom}(a) \to L_1$
  - i.e., actions a with $\texttt{dom}(a) \in L_1, L_2$
  - i.e., actions a with $a \in l_1, l_2$
- so: $\texttt{purge}_{L_1}(\alpha) = l_1 l_2 l_1 l_2$
- so: $\texttt{purge}_{H_1}(\alpha) = \alpha$
- so: $\texttt{purge}_{L_2}(\alpha) = l_1 l_2 l_1 l_2$

P-secure example system

why is the system P-secure?

- need to show: if $\texttt{purge}_L(\alpha_1) = \texttt{purge}_L(\alpha_2)$, and s is a state, then $\texttt{obs}_u(s \cdot \alpha_1) = \texttt{obs}_u(s \cdot \alpha_2)$.
- differently: $\texttt{obs}_L(s \cdot \alpha)$ only depends on:
  - s,
  - $\texttt{purge}_L(\alpha)$
- how can we write $\texttt{obs}_L(s \cdot \alpha)$ as a function of s and $\texttt{purge}_L(\alpha)$?
- $\texttt{obs}_u(s \cdot \alpha) =$
  - if α does not contain any l-action, then this is just $\texttt{obs}_L(s)$,
  - otherwise: observation is the index of the last l-action in α.