Modular Control

- Desired behavior $\underset{\to O(n_1 n_2)}{K} = \overset{\to O(n_1)}{K_1} \cap \overset{\to O(n_2)}{K_2}$
  - buffer should not overflow / underflow: K_1
  - downline m/c has priority of repair: K_2

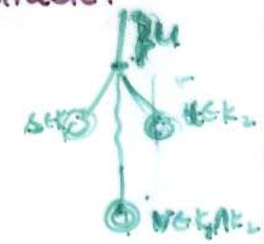
- Modular approach: Design S_i s.t. $G \| S_i$ meets K_i

  Then construct $S := (S_1 \| S_2) \overset{?}{\Rightarrow} G \| S$ meets K

  - Computational advantage: $O(m^2 n_1^2) + O(m^2 n_2^2)$ vs $O(m^2 n_1^2 n_2^2)$.

- Def: K_1, K_2 modular if $\underbrace{[pr(K_1) \cap pr(K_2)] = pr[K_1 \cap K_2]}_{\text{(non-conflicting)}} \Leftrightarrow pr(K_1) \cap pr(K_2) \subseteq pr(K_1 \cap K_2)$
  - clearly this holds for K_1, K_2 prefix-closed.
  - K_1, K_2 share a prefix $\Rightarrow$ they share a trace containing that prefix.

- Lemma: K_1, K_2 modular, K_1, K_2 controllable $\Rightarrow K_1 \cap K_2$ controllable.



$$
\begin{aligned}
pr[K_1 \cap K_2]\Sigma_u \cap L(G) &\subseteq [pr(K_1) \cap pr(K_2)]\Sigma_u \cap L(G) \\
&= [pr(K_1)\Sigma_u \cap L(G)] \cap [pr(K_2)\Sigma_u \cap L(G)] \\
&\overset{\text{controllability}}{\subseteq} pr(K_1) \cap pr(K_2) \underset{\text{modularity}}{=} pr(K_1 \cap K_2)
\end{aligned}
$$

- Modular control is always possible for prefix-closed languages.

- Thm: Given G, K_1, K_2. Suppose exists Σ_u-enabling, non-marking S_i s.t. $L(G \| S_i) = K_i$. Then $S := S_1 \| S_2$ is Σ_u-enabling, non-marking and $L(G \| S) = K_1 \cap K_2$.

- Must have $K_i = \sup PC(K_i)$; otherwise S_i s.t. $L(G \| S_i) = \sup PC(K_i)$

  Thm: $\sup PC(K_1) \cap \sup PC(K_2) = \sup PC(K_1 \cap K_2)$

Modular Control : Non-prefix closed case

- **Thm**: Given G, K_1, K_2. Suppose exists Σ_u-enabling, non-marking, nonblocking S_i s.t. $L_m(G||S_i) = K_i$. Then $S := S_1||S_2$ is Σ_u-enabling, non-marking and $L_m(G||S) = K_1 \cap K_2$. Furthermore, S is non-blocking if and only if K_1, K_2 are modular.

Pf: S_i Σ_u-enabling $\Rightarrow L(G||S_i)$ controllable
$\Rightarrow L(G||S) = L(G||S_1) \cap L(G||S_2)$ controllable
$\Rightarrow S$ Σ_u-enabling

S_i non-marking $\Rightarrow L_m(G||S_i) = L(G||S_i) \cap L_m(G)$
$\Rightarrow L_m(G||S) = L_m(G||S_1) \cap L_m(G||S_2) = [L(G||S_1) \cap L_m(G)] \cap [L(G||S_2) \cap L(G)]$
$= L(G||S) \cap L(G) \Rightarrow S$ non-marking

$L_m(G||S) = L_m(G||S_1) \cap L_m(G||S_2) = K_1 \cap K_2$
$L(G||S) = L(G||S_1) \cap L(G||S_2) = pr(L_m(G||S_1)) \cap pr(L_m(G||S_2)) = pr(K_1) \cap pr(K_2)$

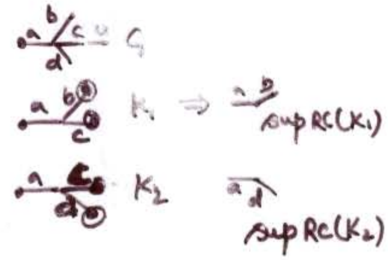
S non-blocking $\Leftrightarrow pr(L_m(G||S)) = L(G||S)$
$\Leftrightarrow pr(K_1 \cap K_2) = pr(K_1) \cap pr(K_2) \Leftrightarrow (K_1, K_2)$ modular.

Complexity: "non-modular" approach: controllability & Relative-closure of $K_1 \cap K_2$
$\Rightarrow O(mn_1n_2) \sim O(n^3)$
"modular" approach: controllability + relative-closure of K_i, modularity of K_1, K
$\Rightarrow O(mn_1) + O(mn_2) + O(n_1n_2) \sim O(n^2)$

- Existence of S_i requires $K_i = supRC(K_i)$; otherwise consider $supRC(K_i)$

Thm: $supRC(K_1)$, $supRC(K_2)$ modular. Then

$$supRC(K_1 \cap K_2) = supRC(K_1) \cap supRC(K_2)$$



Note: K_1, K_2 modular $\not\Rightarrow supRC(K_1), supRC(K_2)$ modular