# Advanced Tools from Modern Cryptography

Lecture 3

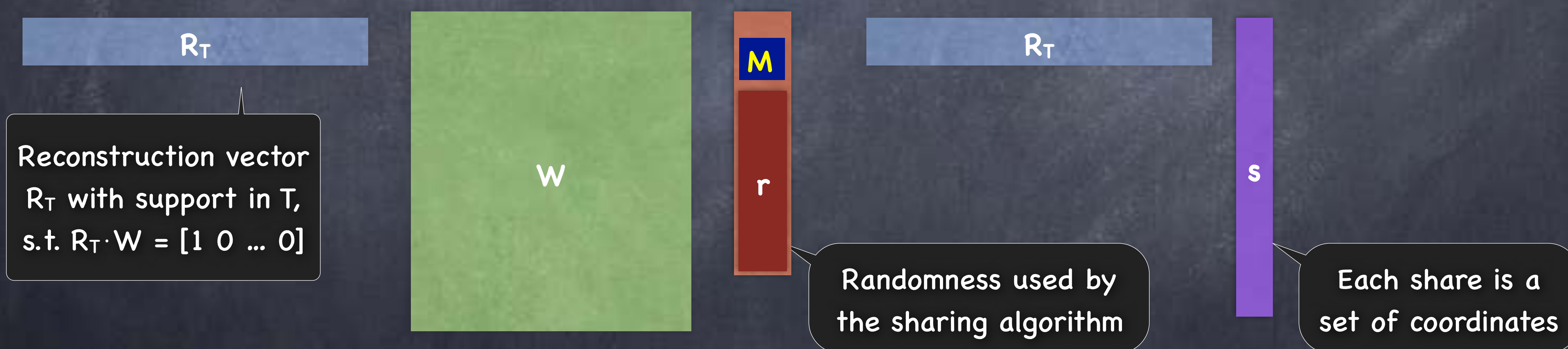
Secret-Sharing (ctd.)

# Secret-Sharing

- Last time
  - (n,t) secret-sharing
    - (n,n) via additive secret-sharing
    - Shamir secret-sharing for general (n,t)
    - Shamir secret-sharing is a linear secret-sharing scheme

# Linear Secret-Sharing

- Linear Secret-Sharing over a field: message and shares are field elements
- Reconstruction by a set $T \subseteq [n]$ : solve $W_T \begin{bmatrix} M \\ r \end{bmatrix} = s_T$ for M



# Linear Secret-Sharing: Computing on Shares

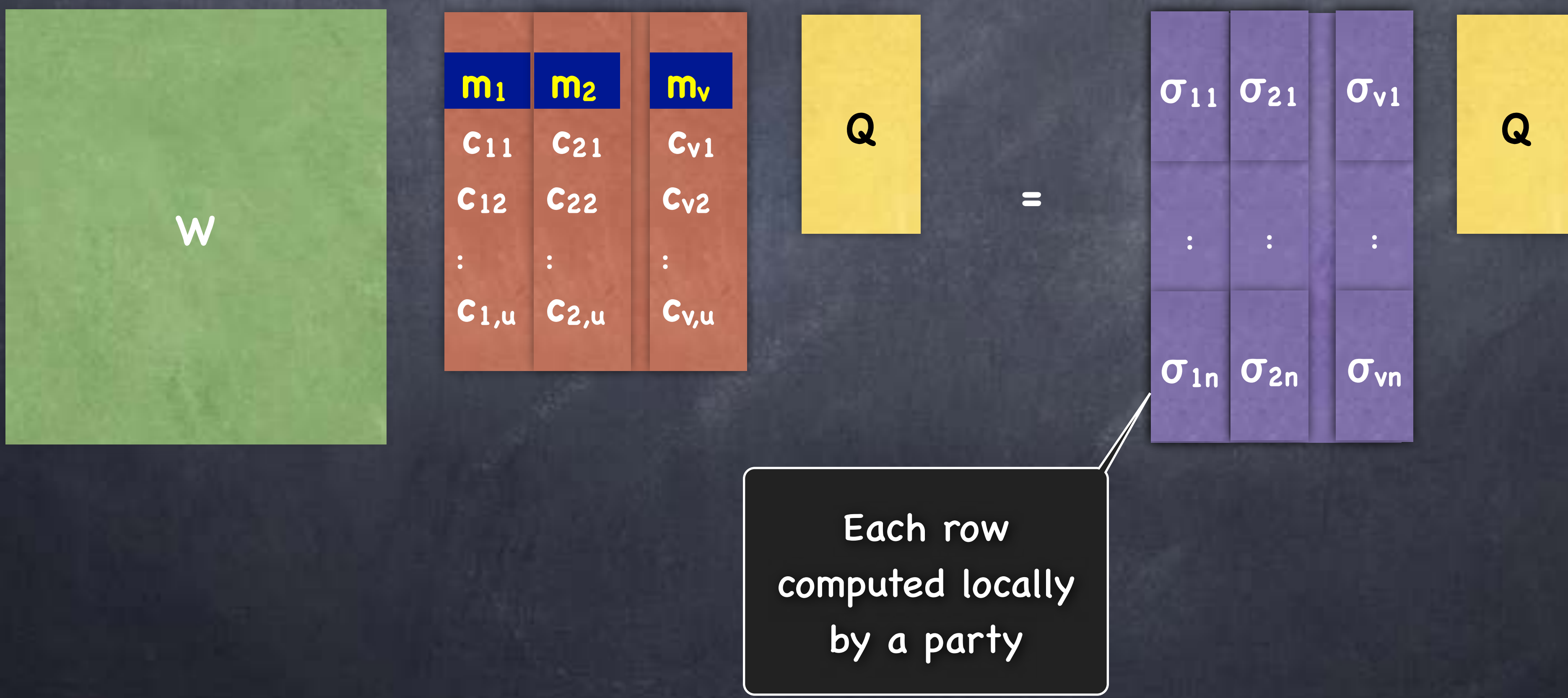
- Suppose two secrets $m_1$ and $m_2$ shared using the same secret-sharing scheme

$$W \begin{bmatrix} m_1 & m_2 \\ c_{11} & c_{21} \\ c_{12} & c_{22} \\ \vdots & \vdots \\ c_{1,u} & c_{2,u} \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{21} \\ \vdots & \vdots \\ \sigma_{1n} & \sigma_{2n} \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix}$$

- Then for any $p,q \in F$, shares of $p \cdot m_1 + q \cdot m_2$ can be computed <u>locally</u> by each party i as $\sigma_i = p \cdot \sigma_{1i} + q \cdot \sigma_{2i}$

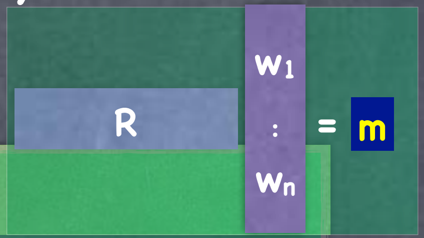
# Linear Secret-Sharing: Computing on Shares

- More generally, can compute shares of any linear transformation

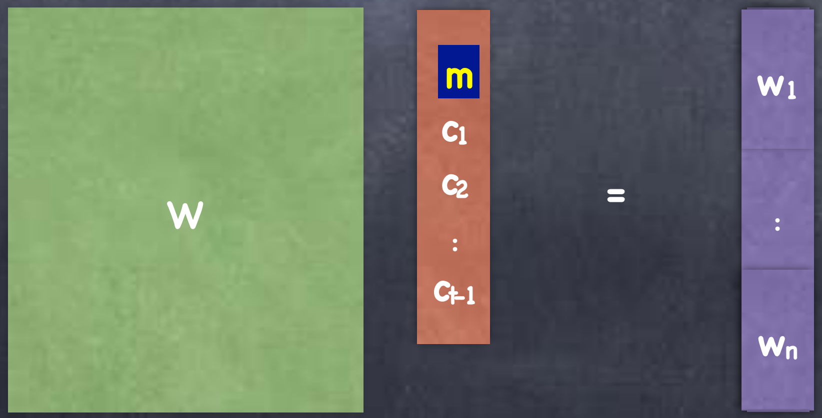


# Switching Schemes

- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"

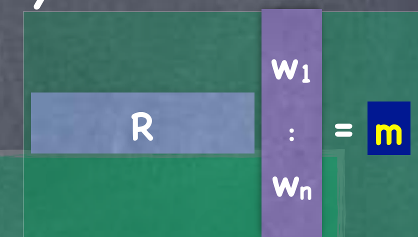



- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each $w_i$ using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share($w_i$)
- Locally each party j reconstructs using scheme W: $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$

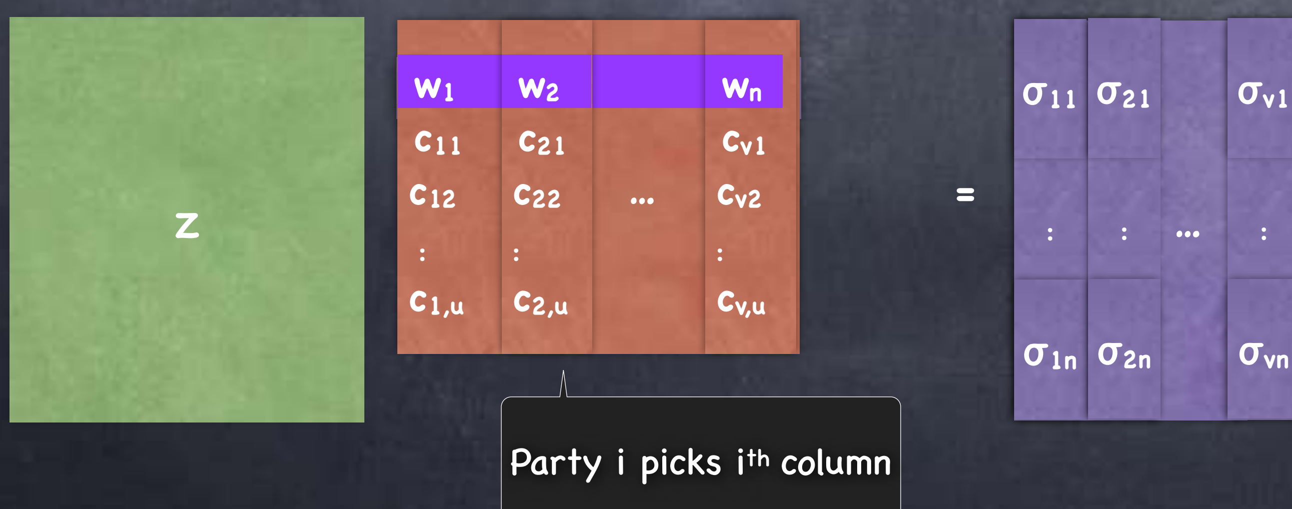


# Switching Schemes

- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"

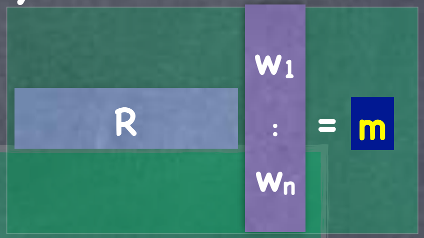



- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each $w_i$ using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share($w_i$)
- Locally each party j reconstructs using scheme W:
  $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$

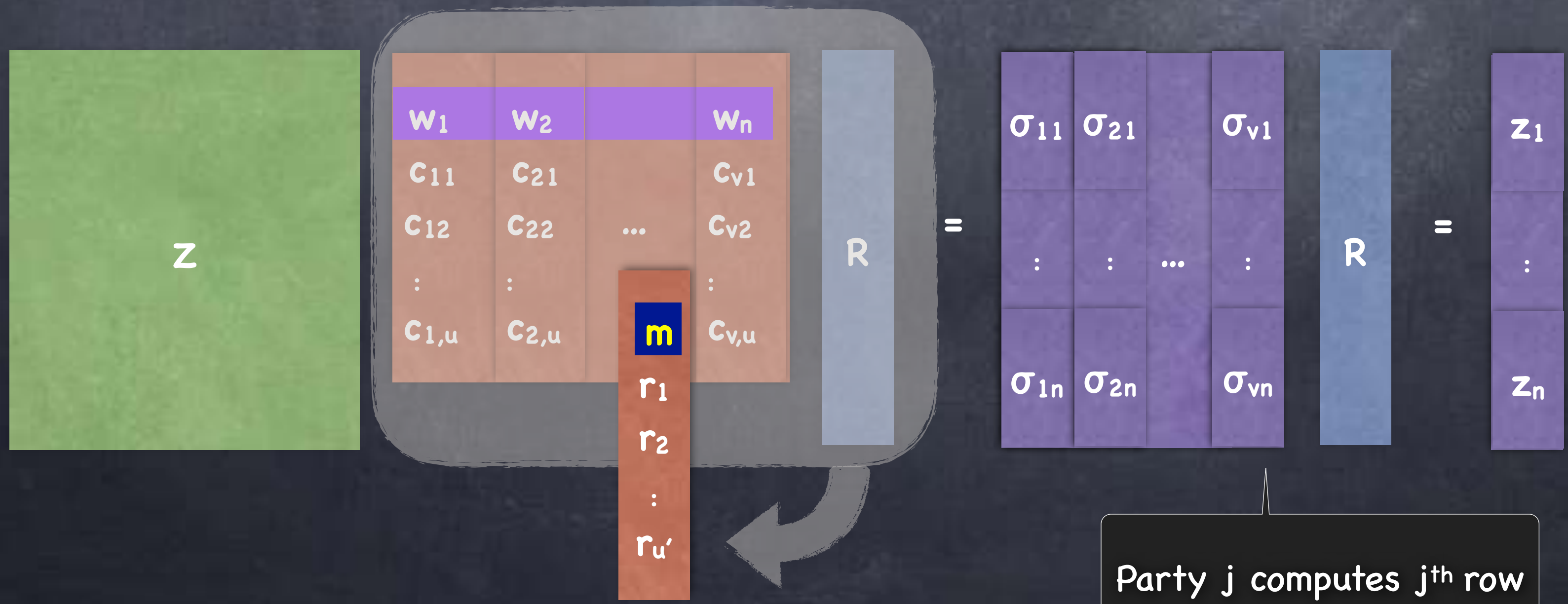


# Switching Schemes

- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"




- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each $w_i$ using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share($w_i$)
- Locally each party j reconstructs using scheme W:
  $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$



# Switching Schemes

- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"

- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each $w_i$ using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share($w_i$)
- Locally each party j reconstructs using scheme W: $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$

- Note that if a set of parties $T \subseteq [n]$ is allowed to learn the secret by either W or Z, then T learns m from either the shares it started with or the ones it ended up with

- Claim: If $T \subseteq [n]$ is not allowed to learn the secret by both W and Z, then T learns nothing about m from this process
  - Exercise

# Efficiency

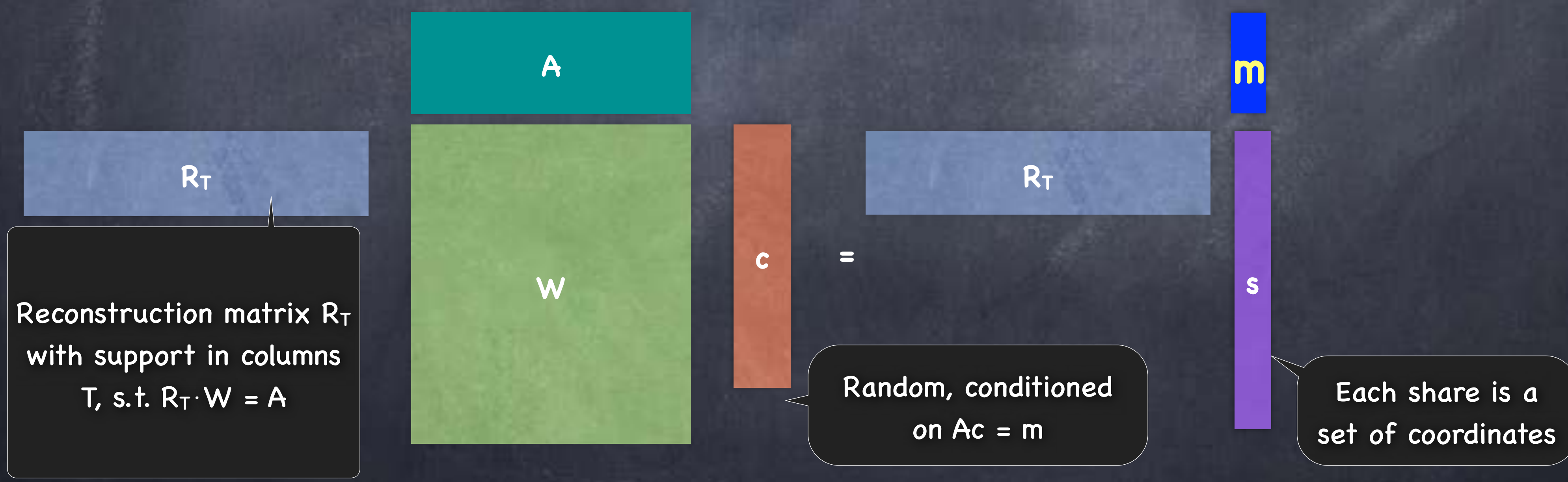
- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - cf. Naïve scheme for arbitrary monotonic access structure $\mathcal{A}$, with "basis" $\mathcal{B}$: if a party is in N sets in $\mathcal{B}$, N basic shares
    - N can be exponential in n (as $\mathcal{B}$ can have exponentially many sets)
  - Share size must be at least as big as the secret: "last share" in a minimal authorized set should contain all the information about the secret
    - <u>Ideal</u>: if all shares are only this big (e.g. Shamir's scheme)
    - Not all access structures have ideal schemes
  - Non-linear schemes can be more efficient than linear schemes

# A More General Formulation

- A generalised access structure consists of a monotonically "increasing" family $\mathcal{A}$ (allowed to learn), and a monotonically "decreasing" family $\mathcal{F}$ (forbidden from learning), with $\mathcal{A} \cap \mathcal{F} = \emptyset$
  - $T \in \mathcal{A} \Rightarrow \forall S \supseteq T,\ S \in \mathcal{A}$. $T \in \mathcal{F} \Rightarrow \forall S \subseteq T,\ S \in \mathcal{F}$.
  - For $T \notin \mathcal{A} \cup \mathcal{F}$, no requirements of secrecy or learning the message
- E.g., Ramp secret-sharing scheme: $\mathcal{A} = \{ S \subseteq [n] \mid |S| \geq t \}$ and $\mathcal{F} = \{ S \subseteq [n] \mid |S| \leq s \}$, where $s < t$
  - When $s = t-1$, a threshold secret-sharing scheme

# Packed Secret-Sharing

- Shamir's scheme can be generalized to a ramp scheme, such that longer secrets can be shared with the same share size
  - $m_j = f(z_j)$ and $s_i = f(a_i)$ where $\{z_1,\ldots,z_k\} \cap \{a_1,\ldots,a_n\} = \emptyset$ and $f$ has degree $t-1$ ($t$ being the reconstruction threshold)
  - Access structure: $\mathcal{A} = \{ S : |S| \geq t \}$ and $\mathcal{F} = \{ S : |S| \leq t\text{-}k \}$




- $T \in \mathcal{A}$ if A spanned by $W_T$, and $T \in \mathcal{F}$ if every row of A independent of $W_T$