Outline

1. Realisability and safe realisability

2. Regular MSCs, MSGs, CFMs

3. Regularity and realisability for MSCs

4. Regularity and realisability for MSGs
   - Communication closedness
Overview

1. Realisability and safe realisability

2. Regular MSCs

3. Regularity and realisability for MSCs

4. Regularity and realisability for MSGs
   - Communication closedness
### Definition (Realisability)

1. MSC $M$ is **realisable** whenever $\{M\} = \mathcal{L}(A)$ for some CFM $A$.
2. A finite set $\{M_1, \ldots, M_n\}$ of MSCs is **realisable** whenever $\{M_1, \ldots, M_n\} = \mathcal{L}(A)$ for some CFM $A$.
3. MSG $G$ is **realisable** whenever $\mathcal{L}(G) = \mathcal{L}(A)$ for some CFM $A$.

### Definition (Safe realisability)

Same as above except that the CFM should be **deadlock-free**.
Approach so far:
The (safe) realisation of a (finite) set of MSCs by a weak CFM is the one where the automaton $A_p$ of process $p$ generates the projections of these MSCs on $p$.

Results so far:
1. Conditions for (safe) realisability for finite sets of MSCs.
2. Checking safe realisability for finite sets of MSCs is in $P$.
3. Checking realisability for finite sets of MSCs is co-NP complete.

Sufficient and necessary conditions
Some remaining questions

- Can similar results be obtained for larger classes of MSGs?
- What happens if we allow synchronisation messages?
  - recall that weak CFMs do not involve synchronisation messages
- How do we obtain a CFM realising an MSG algorithmically?
  - in particular, for local-choice MSGs
- Are there "simple" conditions on MSGs that guarantee realisability?
  - e.g., easily identifiable subsets of (safe) realisable MSGs
Today’s lecture
(Safe) Realisability of a regular set of MSCs.

Or, equivalently: (safe) realisability of a regular set of well-formed words.
(Safe) Realisability of a regular set of MSCs.

Or, equivalently: (safe) realisability of a regular set of well-formed words.

Results:

1. Checking whether a regular language $L$ is well-formed is decidable.
2. For well-formed language $L$:
   
   $L$ is regular iff it is (safely) realisable by a $\forall$-bounded CFM.
3. Checking whether an MSG is regular is undecidable.
4. Every communication-closed MSG is regular.
5. Checking whether an MSG is comm.-closed is coNP-complete.

= same complexity as checking whether a finite set of MSCs is realisable by a weak CFM.
(Safe) Realisability of a regular set of MSCs.
Or, equivalently: (safe) realisability of a regular set of well-formed words.

Results:
1. Checking whether a regular language \( L \) is well-formed is decidable.
2. For well-formed language \( L \):
   \( L \) is regular iff it is (safely) realisable by a \( \forall \)-bounded CFM.
3. Checking whether an MSG is regular is undecidable.
4. Every communication-closed MSG is regular.
5. Checking whether an MSG is comm.-closed is coNP-complete.
6. Checking whether an MSG is locally communication-closed is in P.
Overview

1. Realisability and safe realisability

2. Regular MSCs

3. Regularity and realisability for MSCs

4. Regularity and realisability for MSGs
   - Communication closedness
# Regular MSCs

## Definition (Regular MSCs, MSGs, and CFMs)

1. Let \( M = \{ M_1, \ldots, M_n \} \) with \( n \in \mathbb{N} \cup \{ \infty \} \) is called **regular** if \( \text{Lin}(M) = \bigcup_{i=1}^{n} \text{Lin}(M_i) \) is a regular word language over \( \text{Act}^* \).  
2. MSG \( G \) is **regular** if \( \text{Lin}(G) \) is a regular word language over \( \text{Act}^* \).  
3. CFM \( A \) is **regular** if \( \text{Lin}(A) \) is a regular word language over \( \text{Act}^* \).  

Here, \( \text{Act} \) is the set of actions in \( M, G, \) and \( A \), respectively.

\[
\text{Lin}(M) = a^*b^* 
\]

\( \varepsilon \) is regular  
\( \varnothing \) is regular  
\( \forall a \in \Sigma. \{ a \} \) is regular  
if \( A \) and \( B \) are regular, then \( A + B \) (or \( A \cup B \)) is regular.
Definition (Regular MSCs, MSGs, and CFMs)

1. $\mathcal{M} = \{M_1, \ldots, M_n\}$ with $n \in \mathbb{N} \cup \{\infty\}$ is called regular if $\text{Lin}(\mathcal{M}) = \bigcup_{i=1}^{n} \text{Lin}(M_i)$ is a regular word language over $\text{Act}^*$.

2. MSG $G$ is regular if $\text{Lin}(G)$ is a regular word language over $\text{Act}^*$.

3. CFM $A$ is regular if $\text{Lin}(A)$ is a regular word language over $\text{Act}^*$.

Here, $\text{Act}$ is the set of actions in $\mathcal{M}$, $G$, and $A$, respectively.

Lemma:

Every $\forall$-bounded CFM is regular.

Why?

$\exists$ has finitely many configurations.
its configuration graph is a finite-state automaton.
Examples

A.

Claim: this MSG is regular.

\[ \text{Lin}(A) = \left( ! (p, q, a) ? (q, p, a) + ! (q, p, b) ? (p, q, b) \right) + \]

= \left( ! a ? a ! b ? b \right) ^ +

is a regular language

communication closed

B.

Claim: this MSG is not regular.

let \( ! (p, q, a) \) be “[” (open bracket)

\( ? (q, p, a) \) be “]” (closing bracket)

Then \( L(G') = \text{Dyck language} = \text{not regular} \)

= language of words that contained a balanced set of brackets “[“ and “]”

e.g. [[ ]] or [[[[]]]] etc.

not [[ ]]
Claim: this CFM is regular, as it is \( \forall 3 \) bounded.
Theorem

The decision problem “is a regular language $L \subseteq \text{Act}^*$ well-formed”?—that is, does regular $L$ represent a set of MSCs?—is decidable.

Proof.

Since $L$ is regular, there exists a minimal DFA $A = (S, \text{Act}, s_0, \delta, F')$ with $L(A) = L$. Consider the productive states in this DFA, i.e., all states from which some state in $F$ can be reached. We label every productive state $s$ with a channel-capacity function $K_s : \text{Ch} \rightarrow \mathbb{N}$ such that four constraints (cf. next slide) are fulfilled. Then: $L$ is well-formed iff each productive state in the DFA $A$ can be labelled with $K_s$ satisfying these constraints. In fact, if a state-labelling violates any of these constraints, it is due to a word that is not well-formed.
Constraints on state-labelling

1. \( s \in F \cup \{s_0\} \), implies \( K_s((p, q)) = 0 \) for every channel \((p, q)\).

2. \( \delta(s, !(p, q, a)) = s' \) implies

\[
K_{s'}(c) = \begin{cases} 
K_s(c) + 1 & \text{if } c = (p, q) \\
K_s(c) & \text{otherwise.}
\end{cases}
\]
Constraints on state-labelling

1. \( s \in F \cup \{s_0\} \), implies \( K_s((p, q)) = 0 \) for every channel \((p, q)\).

2. \( \delta(s, !(p, q, a)) = s' \) implies

\[
K_{s'}(c) = \begin{cases} 
K_s(c) + 1 & \text{if } c = (p, q) \\
K_s(c) & \text{otherwise.}
\end{cases}
\]

3. \( \delta(s, ?(p, q, a)) = s' \) implies \( K_s((q, p)) > 0 \) and

\[
K_{s'}(c) = \begin{cases} 
K_s(c) - 1 & \text{if } c = (q, p) \\
K_s(c) & \text{otherwise.}
\end{cases}
\]
Constraints on state-labelling

1. \( s \in F \cup \{s_0\} \), implies \( K_s((p, q)) = 0 \) for every channel \((p, q)\).

2. \( \delta(s, !(p, q, a)) = s' \) implies

   \[
   K_{s'}(c) = \begin{cases} 
   K_s(c) + 1 & \text{if } c = (p, q) \\
   K_s(c) & \text{otherwise.}
   \end{cases}
   \]

3. \( \delta(s, ?(p, q, a)) = s' \) implies \( K_s((q, p)) > 0 \) and

   \[
   K_{s'}(c) = \begin{cases} 
   K_s(c) - 1 & \text{if } c = (q, p) \\
   K_s(c) & \text{otherwise.}
   \end{cases}
   \]

4. \( \delta(s, \alpha) = s_1 \) and \( \delta(s_1, \beta) = s_2 \) with \( \alpha \in \text{Act}_p \) and \( \beta \in \text{Act}_q \), \( p \neq q \), implies

   not \((\alpha = !(p, q, a) \text{ and } \beta = ?(q, p, a))\), or \( K_s((p, q)) > 0 \) implies \( \delta(s, \beta) = s'_1 \) and \( \delta(s'_1, \alpha) = s_2 \) for some \( s'_1 \in S \).
Constraints on state-labelling

1. $s \in F \cup \{s_0\}$, implies $K_s((p, q)) = 0$ for every channel $(p, q)$.

2. $\delta(s, !(p, q, a)) = s'$ implies

$$K_{s'}(c) = \begin{cases} 
K_s(c) + 1 & \text{if } c = (p, q) \\
K_s(c) & \text{otherwise.}
\end{cases}$$

3. $\delta(s, ?(p, q, a)) = s'$ implies $K_s((q, p)) > 0$ and

$$K_{s'}(c) = \begin{cases} 
K_s(c) - 1 & \text{if } c = (q, p) \\
K_s(c) & \text{otherwise.}
\end{cases}$$

4. $\delta(s, \alpha) = s_1$ and $\delta(s_1, \beta) = s_2$ with $\alpha \in Act_p$ and $\beta \in Act_q$, $p \neq q$, implies

not $(\alpha = !(p, q, a)$ and $\beta = ?(q, p, a))$, or $K_s((p, q)) > 0$

implies $\delta(s, \beta) = s'_1$ and $\delta(s'_1, \alpha) = s_2$ for some $s'_1 \in S$.

These constraints can be checked in linear time in the size of relation $\delta$.
Ap \rightarrow 1 \xrightarrow{a} 2 \xrightarrow{1a} 3

Ag \rightarrow A \xrightarrow{b} B \xrightarrow{?a} C

configurations

1A00 \rightarrow \begin{cases} 
A_p \text{ is in state 1} \\
Ag \text{ is in state A} \\
K(\text{Ag}) = K(\text{a}, \text{p}) = 0
\end{cases}

configuration graph is defined by the following rules

1A00
\begin{align*}
\rightarrow & 2A10 \\
\rightarrow & 1B01
\end{align*}

2An nm
\begin{align*}
\rightarrow & 3A n+1 m \\
\rightarrow & 2B n m+1 \\
\rightarrow & 2C n-1 m \quad \text{if } n > 0
\end{align*}

3 A nm
\begin{align*}
\rightarrow & 2A n m-1 \quad \text{if } m > 0 \\
\rightarrow & 3B n m+1 \\
\rightarrow & 3C n-1 m \quad \text{if } n > 0
\end{align*}

3 B nm
\begin{align*}
\rightarrow & 2B n m-1 \quad \text{if } m > 0 \\
\rightarrow & 3A n-1 m \quad \text{if } n > 0
\end{align*}
3 \text{Cn} m \rightarrow 2 \text{Cn} m \rightarrow 1 \text{Bn} m \rightarrow 1 \text{An} n-1 m \text{ if } n>0

?\alpha \rightarrow \text{3C10}

!\alpha \rightarrow \text{2C00}

?\alpha \rightarrow \text{2A10}

!\alpha \rightarrow \text{3A20}

!\beta \rightarrow \text{2B11}

!\alpha \rightarrow \text{1A00}

I\!\alpha \rightarrow \text{1B01}
\[ \alpha, \beta \text{ are not matched or } k(p,q) > 0 \]

DFA fulfills all constraints \( \Rightarrow \) well-formed
Boundedness and regularity

Definition (\(B\)-bounded words)

Let \(B \in \mathbb{N}\) and \(B > 0\). A word \(w \in \text{Act}^*\) is called \(B\)-bounded if for any prefix \(u\) of \(w\) and any channel \((p, q) \in Ch\):

\[
0 \leq \sum_{a \in C} |u|!(p,q,a) - \sum_{a \in C} |u|?(q,p,a) \leq B
\]

Corollary:

For any regular, well-formed language \(L\), there exists \(B \in \mathbb{N}\) and \(B > 0\) such that every \(w \in L\) is \(B\)-bounded.

Proof.

The bound \(B\) is the largest value attained by the channel-capacity functions assigned to productive states in the proof of the previous theorem.
Overview

1. Realisability and safe realisability

2. Regular MSCs

3. Regularity and realisability for MSCs

4. Regularity and realisability for MSGs
   - Communication closedness
Regularity and realisability

Theorem: [Henriksen et al., 2005], [Baudru & Morin, 2007]
For well-formed $L$, the following four statements are equivalent:

1. $L$ is regular.
2. $L$ is realisable by a $\forall$-bounded CFM.  
3. $L$ is realisable by a deterministic $\forall$-bounded CFM.
4. $L$ is safe realisable by a $\forall$-bounded CFM.  

\{ cf. lecture 10 \} \{ cf. lecture 11 \}
Regularity and realisability

Theorem: \cite{Henriksen et al., 2005}, \cite{Baudru & Morin, 2007} 

For well-formed $L$, the following four statements are equivalent:

1. $L$ is regular.
2. $L$ is realisable by a $\forall$-bounded CFM.
3. $L$ is realisable by a deterministic $\forall$-bounded CFM.
4. $L$ is safe realisable by a $\forall$-bounded CFM.

Lemma:

The maximal size of the CFM realising $L$ is such that for each process $p$, the number $|Q_p|$ of states of local automaton $A_p$ is:

1. double exponential in the bound $B$ and $k^2$, where $k = |\mathcal{P}|$, and
2. exponential in $m \log m$ where $m$ is the size of the minimal DFA for $L$. 
Overview

1. Realisability and safe realisability
2. Regular MSCs
3. Regularity and realisability for MSCs
4. Regularity and realisability for MSGs
   - Communication closedness
Regularity for MSGs is undecidable

Theorem

[Henriksen et. al, 2005]

The decision problem “is MSG $G$ regular“? is **undecidable**

Proof

Outside the scope of this lecture.
Towards structural conditions for regular MSGs

- MSG $G$ is regular if $Lin(G)$ is a regular language

- Regularity yields deterministic, or safe, but bounded CFMs

- But, “is MSG $G$ regular“? is unfortunately **undecidable**

- Is it possible to impose structural conditions on MSGs that guarantee regularity?

  MSG $G$ satisfies the structural conditions implies $G$ is realisable
Towards structural conditions for regular MSGs

- MSG $G$ is regular if $Lin(G)$ is a regular language

- Regularity yields deterministic, or safe, but bounded CFMs

- But, “is MSG $G$ regular?” is unfortunately **undecidable**

- Is it possible to impose **structural** conditions on MSGs that guarantee regularity?

- **Yes we can.** For instance, by constraining:
  1. the communication structure of the MSCs in loops of $G$, or
  2. the structure of expressions describing the MSCs in $G$
Definition (Communication graph)

The communication graph of the MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, <)$ is the directed graph $(V, \rightarrow)$ with:

- $V = \mathcal{P} \setminus \{ p \in \mathcal{P} \mid E_p = \emptyset \}$, the set of active processes
- $(p, q) \in \rightarrow$ if and only if $\mathcal{L}(e) = !(p, q, a)$ for some $e \in E$ and $a \in \mathcal{C}$

Example

an example MSC
Let $G = (V, \rightarrow)$ be a directed graph.

**Strongly connected component**

- $T \subseteq V$ is **strongly connected** if for every $v, w \in T$, vertices $v$ and $w$ are mutually reachable (via $\rightarrow$) from each other.
Let $G = (V, \rightarrow)$ be a directed graph.

**Strongly connected component**

- $T \subseteq V$ is strongly connected if for every $v, w \in T$, vertices $v$ and $w$ are mutually reachable (via $\rightarrow$) from each other.

- $T$ is a strongly connected component (SCC) of $G$ if $T$ is strongly connected and $T$ is not properly contained in another SCC.

Determining the SCCs of a digraph can be done in linear time in the size of $V$ and $\rightarrow$.

*E.g. depth-first algorithm*
Communication closedness
A loop is \textit{simple} if it visits a vertex at most once, except for the start- and end-vertex which are visited twice.
A loop is **simple** if it visits a vertex at most once, except for the start- and end-vertex which are visited twice.

**Definition (Communication closedness)**

MSG $G$ is **communication-closed** if for every simple loop $\pi = v_1 v_2 \ldots v_n$ (with $v_1 = v_n$) in $G$, the communication graph of the MSC $M(\pi) = \lambda(v_1) \bullet \lambda(v_2) \bullet \ldots \bullet \lambda(v_n)$ is strongly connected.

**Example**

On the black board.
**Example**

MSG \( G \):

![Graph drawing]

A single loop (which is simple): \( v_1 \), \( v_2 \), \( v_3 \), \( v_1 = \pi \)

**Communication graph of** \( M(\pi) \): \( \lambda(v_1) \cdot \lambda(v_2) \cdot \lambda(v_3) \)

Thus \( G \) is communication closed.
Communication graph of \( g' \):

\[ \Pi = u_1 u_2 u_1 \]

\[ M(\Pi) : \]

\[ \begin{array}{c}
1 & \rightarrow & 2 \\
\text{not strongly connected} \\
3 & \rightarrow & 4 \\
\end{array} \]

Thus, \( g' \) is not communication closed.

\( L(\Pi) \) is not regular. To see this, consider

\[ \text{Lin}(g') \cap \{ !a, !c \} = \]

\[ \{ \sigma \in \{ !a, !c \}^* \mid \# !a \sigma = \# !c \sigma > 0 \} \]

is not regular. As regular languages are closed under projections, \( \text{Lin}(g') \) is not regular.
Theorem:
Every communication-closed MSG $G$ is regular.

Example on the black board.

Note:
The converse does not hold (cf. next slide).
Communication-closed vs. regularity

Communication-closedness is not a necessary condition for regularity:

\[(!a?a!b?b)^+ \text{ interleaved } (!a?a!b?b)^+\]

\[
\begin{align*}
\text{between } p_1 \text{ and } p_2 \\
\text{between } p_3 \text{ and } p_4
\end{align*}
\]

\[G:\]

MSG $G$ is not communication-closed, but $Lin(G)$ is regular.
Theorem: [Genest et. al, 2006]

The decision problem “is MSG G communication closed?” is co-NP complete.

equally hard as checking whether a finite set of MSCs is realisable by a weak CFM (cf. lecture 10)
Theorem: [Genest et al, 2006]

The decision problem “is MSG $G$ communication closed?” is co-NP complete.

Proof

1. Membership in co-NP can be proven in a standard way: guess a sub-graph of $G$, check in polynomial time whether this sub-graph has a loop passing through all its vertices, and check whether its communication graph is not strongly connected. \( \text{(in poly time)} \)

2. Co-NP hardness can be shown by a reduction from the 3-SAT problem.
Theorem: Checking whether MSG $G$ is comm.-closed is conp-hard.

Proof: Polynomial reduction from the 3SAT-problem.

3SAT: consider the Boolean formula

$$\phi = C_1 \land \ldots \land C_m$$

over the variables \{ $x_1, \ldots, x_n$ \} such that clause

$$C_j = e_j^1 \lor e_j^2 \lor e_j^3$$

literals equals $x_k$ or $\overline{x_k}$ for some $k \in \{1, \ldots, n\}$

$\phi$ is satisfiable if \exists valuation for $x_1$ through $x_n$.

for every $m$, $C_m$ is true.

Fact: 3SAT is NP-complete. Its complement is also NP-complete, thus 3SAT is also coNP-complete.

Reduction:

3SAT-formula

$$\phi = C_1 \land \ldots \land C_m$$

over \{ $x_1, \ldots, x_n$ \}

such that $\phi$ is satisfiable iff $G$ has a simple loop that is not strongly connected.

$G$ is not comm. closed.
The structure of MSG G is as follows: n variables

\[ N_0 \quad \rightarrow \quad N_1 \quad \rightarrow \quad N_2 \quad \rightarrow \quad \ldots \quad \rightarrow \quad N_{n-1} \quad \rightarrow \quad N_n \]

Two vertices \( x_2 \) for \( x_1 \)

\[ \text{if } x_i = \text{true then traverse vertex } NT_i \]
\[ \text{if } x_i = \text{false } (\overline{x_i} = \text{true}) \text{ then traverse } NF_i \]

\[ \phi = c_1 \wedge \ldots \wedge c_m \quad c_j = l_{j1}^1 \lor l_{j2}^2 \lor l_{j3}^3 \]

Processes of the MSCs:

\[ \{ P_0, P_1^1, P_1^2, P_1^3, \ldots, P_m^1, P_m^2, P_m^3, P_{m+1} \} \]

correspond to literals

\[ l_1^1, l_1^2, l_1^3 \]

\[ |P| = 3 \times \#\text{clauses} + 2 \]
\[ \lambda(N_0) = \text{local actions} \]

Template MSCs: \( LT^1_j \), \( LT^2_j \), \( LT^3_j \) (time)

\( LT^3_j \) :: \( P_0 \) \( P_j \)

Also have templates \( LF^1_j \), \( LF^2_j \), \( LF^3_j \)
\( d(NT_i) = \) the concatenation of all template MSCs \( LT_{j}^{k} \) with \( l_{j}^{k} = x_{j} \) 
+ all template MSCs \( LF_{j}^{k} \) with \( l_{j}^{k} = \overline{x_{j}} \) 

\( (k = 1,2,3) \)

\( d(NF_{i}) = \) concatenate:

\( LT_{j}^{k} \) with \( l_{j}^{k} = x_{j} \)
and \( LF_{j}^{k} \) with \( l_{j}^{k} = \overline{x_{j}} \)
Example

\[ \phi = (x_1 \lor \overline{x_2} \lor x_3) \land (x_1 \lor x_2 \lor \overline{x_3}) \lor (x_1 \lor \overline{x_2} \lor \overline{x_3}) \]

\[ \lambda (\text{NT}_2) = \lambda \left( \begin{array}{c}
\overline{\text{LT}}_2 \\overline{\text{LT}}_3 \\overline{\text{LF}}_1
\end{array} \right) \]
Claim: processes $p_a$ and $p_m$ are connected in the communication graph of MSG $G$ iff there exists a clause in $\phi$ for which all literals are false.
Asynchronous iteration

**Definition**

For $M_1, M_2 \subseteq M$ sets of MSCs, let:

$$M_1 \cdot M_2 = \{ M_1 \cdot M_2 \mid M_1 \in M_1, M_2 \in M_2 \}$$

For $M \subseteq M$ let

$$M^i = \begin{cases} \{ M_\epsilon \} & \text{if } i=0, \text{ where } M_\epsilon \text{ denotes the empty MSC} \\ M \cdot M^{i-1} & \text{if } i > 0 \end{cases}$$

The asynchronous iteration of $M$ is now defined by:

$$M^* = \bigcup_{i \geq 0} M^i.$$
Definition (Finitely generated)

Set of MSCs $\mathcal{M}$ is finitely generated if there is a finite set of MSCs $\hat{\mathcal{M}}$ such that $\mathcal{M} \subseteq \hat{\mathcal{M}}^\ast$.

Remarks:

1. Each set of MSCs defined by an MSG $G$ is finitely generated.
2. Not every regular well-formed language is finitely generated.
3. Not every finitely generated set of MSCs is regular.
4. It is decidable to check whether a set of MSCs is finitely generated.
Characterisation of communication-closedness

Theorem: [Henriksen et. al, 2005]

Let $\mathcal{M}$ be a (possibly infinite) set of MSCs. Then:

$\mathcal{M}$ is finitely generated and regular

iff

$\mathcal{M} = \mathcal{L}(G)$ for some communication-closed MSG $G$. 
Definition (Local communication-closedness)

MSG $G$ is locally communication-closed if for each edge $(v, v')$ in $G$, the MSCs $\lambda(v)$, $\lambda(v')$, and $\lambda(v) \cdot \lambda(v')$ all have weakly connected communication graphs.
Local communication-closedness

Definition (Local communication-closedness)

MSG $G$ is **locally** communication-closed if for each edge $(v, v')$ in $G$, the MSCs $\lambda(v)$, $\lambda(v')$, and $\lambda(v) \bullet \lambda(v')$ all have **weakly** connected communication graphs.

Notes:

1. A directed graph is weakly connected if its induced **undirected** graph (obtained by ignoring the directions of edges) is strongly connected.

2. Checking whether MSG $G$ is locally communication-closed can be done in linear time.
Locally communication-closed MSGs are realisable

Theorem:

Every locally communication-closed MSG $G$ is realisable by a CFM $A$ of size $m^O(|\mathcal{P}|)$ where $m$ is the number of vertices in $G$. 

[Genest et al., 2006]