Tema 2 - AA

KClique $\le_p$ SAT

$f \le_p g$ dacă $\exists T : \Sigma^* \to \Sigma^*$, T - transformare ce poate fi rezolvată în timpi polinomiali astfel încât $\forall w \in \Sigma^*$,

$$f(w) = 1 \iff g(T(w)) = 1$$

Variabilele folosite pentru SAT:

- $a_{\alpha\beta}$, unde
  - $\alpha \in \overline{1,N}$ = nr. de ordin al nodului
  - $\beta \in \overline{1,k}$ = nr. nodului în clică
- $a_{\alpha\beta} = 1$, dacă nodul α ocupă poziția β în S.

S = submulțimea cu nodurile din clică

În realizarea clicii am folosit următoarele clauze:

I $\forall \beta \in \overline{1,k} \Rightarrow (a_{1\beta} \vee a_{2\beta} \vee a_{3\beta} \vee \dots \vee a_{i\beta})$

→ putem trage concluzia că dacă luăm pe rând fiecare poziție ~~nod~~, cel puțin un ~~nod~~ nod se află ~~...~~ acolo, deci propoziția devine imediat adevărată

Dem.:

$(a_{11} \vee a_{21} \vee \dots \vee a_{N1})$

$(a_{12} \vee a_{22} \vee \dots \vee a_{N2})$

$(\dots)$

$(a_{1k} \vee a_{2k} \vee \dots \vee a_{Nk})$

II Un nod nu poate fi simultan în mai multe locuri.

Ex.: $(a_{\beta i} \vee a_{\beta j})$ cu condițiile: $\rightarrow \beta \in \overline{1,N}$; $\rightarrow i,j \in \overline{1,K}$; $\rightarrow i \neq j$;

III Nu pot exista 2 noduri, care să nu fie conectate

→ construim clauze care nu există ~~[illegible]~~ capetele neconectate de muchii în clică

→ pt. construirea lor, luăm toate muchiile care nu există în mulțimea muchiilor

(De menționat că ambele probleme sunt de tipul NP-complet, rezultând o singură reducere de la SAT la kClique. Și de asemenea, considerăm că S = submulțime a grafului $G(V,E)$ cu N noduri, care realizează o clică de lungime K.)

Reducerea se rezolvă în timp polinomial, datorită faptului că K este fix. Dacă nu era un K setat, algoritmul s-ar fi rezolvat în timp exponențial.

Pt. clauzele de tipul I, am 2 for-uri, unul în altul, de unde rezultă complexitatea $O(n^2)$.

Pt. clauzele de tipul II, am 3 for-uri, unul în altul, de unde rezultă complexitatea $O(n^3)$.

Pt. clauzele de tipul III, am 4 for-uri, unul în altul, de unde rezultă complexitatea $O(n^4)$.

Clauzele se execută secvențial => complexitățile se adună, de unde rezultă complexitatea $O(n^4)$.