Pina Kolling
piko0011

Summary: Lecture 8

Summary for the chapters *9.3* and *9.4*. [2, 1]

Undirected graph

An undirected graph G is a pair of sets (V, E) where V is the finite set of nodes and E is a set of unordered pairs in V that are symmetric:

$$\forall\, i, j \in E, i \neq j :\ (i, j) \in E \Rightarrow (j, i) \in E$$

IndependentSet

IndependentSet

Input: An undirected Graph $G = (V, E)$ and a number k.
Question: Is there a set $I \subseteq V$ of $k = |I|$ nodes with no edges in between? (IndependentSet)

3SAT

Like the Sat problem, 3Sat is determining the satisfiability of a formula in CNF where each clause is limited to at most three literals.

IndependentSet is NP-complete.

Proof idea:

- triangle construction: any independent set can contain at most one node of the triangle

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

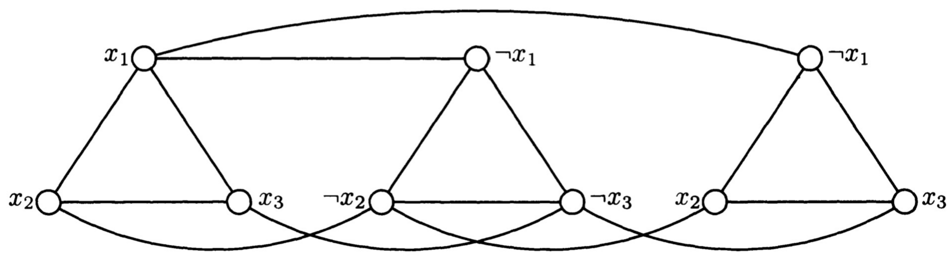

Figure 1: Graph with triangles [2]

- consider only graphs whose nodes can be partitioned in m disjoint triangles
  $\rightarrow$ independent set can contain at moast m nodes (one from each triangle)
- reduction from 3Sat to IndependentSet
- construct graph of formula ϕ:
  - each literal as a node
  - clauses as triangles
  - edges between nodes in different triangles if they correspond to the same literal (negated)
  - $K = m$ (m clauses)

- given: instance ϕ of 3SAT with m clauses $C_1, ..., C_m$
- each clause $C_i = (\alpha_{i1} \vee \alpha_{i2} \vee \alpha_{i3})$ (with α as boolean variables or negation of those)
- reduction R constructs a graph: $R(\phi) = (G, K)$ where $K = m$ and $G = (V, E)$
- nodes $V = \{v_{ij} : \ i = 1, ..., m; j = 1, 2, 3\}$
  nodes for each of the m clauses (i) for each of the 3 literals (j)
- edges $E = \{[v_{ij}, v_{ik}] : \ i = 1, ..., m; j \neq k\} \cup \{[v_{ij}, v_{lk}] : \ i \neq l, \alpha ij = \neg\alpha_{lk}\}$
  edges between the nodes in one clause (triangle edges)
  edges between nodes with the same corresponding literal, but negated

- there is an indepentent set I of K nodes in G only if ϕ is satisfiable
- I must contain a node from each triangle
- negated literals are connected: I cannot contain a literal and its negation
- I is a truth assignment of ϕ:
  - true literals: nodes in I
  - one true literal per clause

HamiltonPath is NP-complete

HamiltonPath

A HAMILTONPATH is a path in a graph that visits each node exactly once.

HAMILTONPATH is NP-complete.

Proof idea:

- reduction from 3SAT to HAMILTONPATH
- given: formula ϕ in CNF with n variales $x_1, ..., x_n$ and m clauses $C_1, ..., C_m$ with each 3 variables
- construct a graph $R(\phi)$ that has a hamilton path only if ϕ is satisfiable:
- boolean variables:
  - choice between true and false
  - all occurences of x must have the same value (and $\neg x$ the opposite)
  - use *choice* gadget (like flip flop)
- XOR:

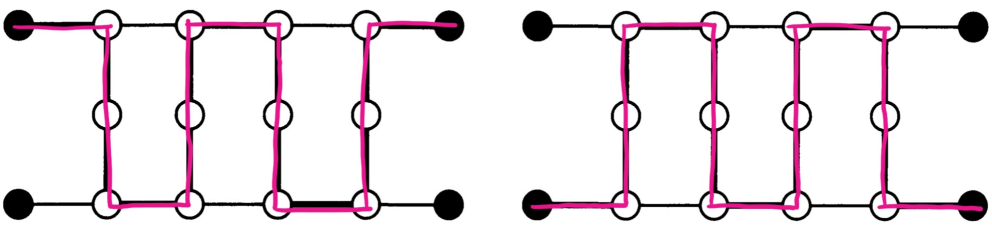

Figure 2: XOR subgraph from the book [2] with the relevant edges marked additionally

  - use *consistency* gadget
  - because of hamilton path: there are only two ways to traverse through this sub graph (as shown above)
  - leads to exclusive or (XOR)

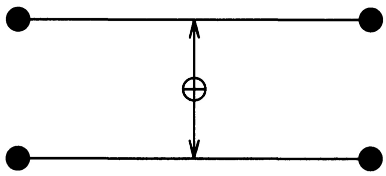

Figure 3: XOR connecting two independent edges (*consistency* gadget) [2]

- clauses:
  - triangles for clause construction
  - one side for each literal
  - if literal is false: hamilton path traveres triangle side
  - at least one literal need to be true: else all three edges of triangle will be traversed and this is not a hamilton path
- put everything together as graph G:
  - G has n copies of the *choice* gadget as a chain (one for each variable)

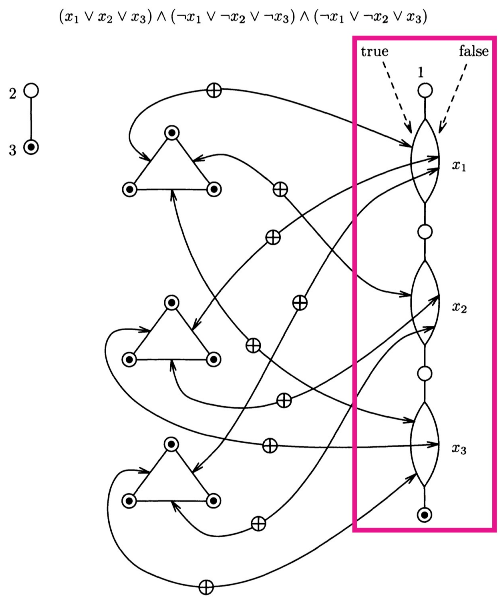

Figure 4: *Choice* gadgets marked in graph from the book [2]

- G has m triangles (one for each clause) with edges for each clause in the triangle

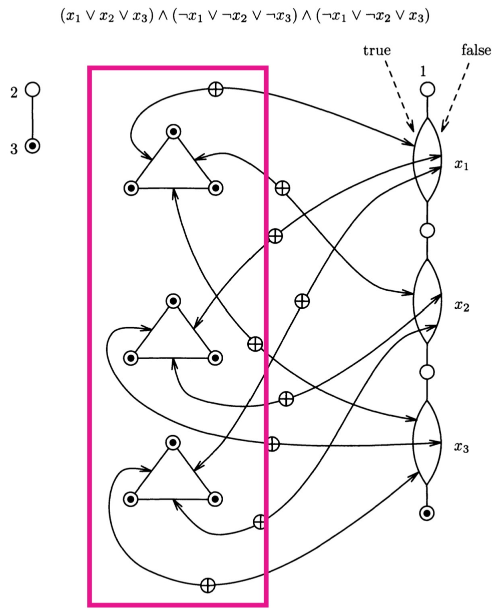

Figure 5: Clauses marked in graph from the book [2]

- finally all $3m$ nodes of the triangles, the last node of the chain of *choice* gadgets and a new node 3 are connected with all possible edges
- a single node 2 is connectes to the node 3

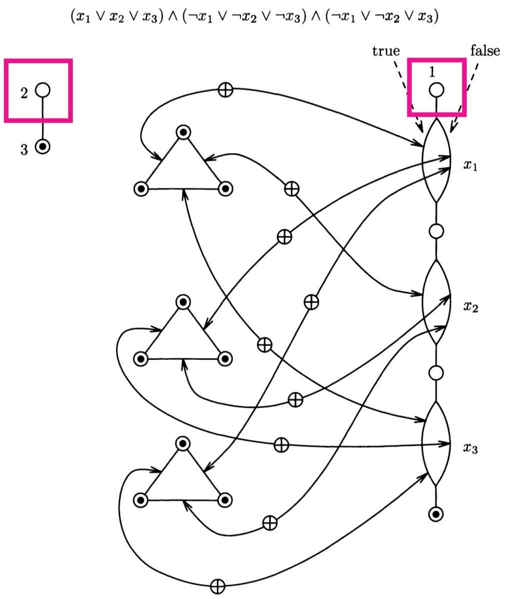

Figure 6: Nodes 1 and 2 marked (start and end node) in graph from the book [2]

- graph has a hamilton path only if ϕ has a satisfying truth assignment
- for hamilton path: start node is node 1 and end node is node 2
- from node 1 it must traverse one of the parallel edges of the *choice* gadget for the first variable
- exclusive ors must be traversed
- whole chain of *choice* gadgets will be traversed
  $\rightarrow$ in this way a truth assignment T is created

- then the triangles are traversed and it ends up in node 2 if there is a hamilton path and ϕ is satisfyable

TSP(D)

TSP(D)

TSP(D) is a decision version of TSP.
Input: A $n \times n$ distance matrix and a bound $B \in \mathbb{N}$
Question: Is there a round tour of length $\leq B$ that visits all *cities*?

TSP(D) is NP-complete.

Proof idea:

- budget of nodes is $B = |V| + 1$

TODO
Questions:

Knapsack

Knapsack

Knapsack is NP-complete.

- filled in in one dimensional array onthe board
-

TODO
Questions:

References

[1] Martin Berglund. *Lecture notes in Computational Complexity.*

[2] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley Publishing Company, 1994.