AUTOMATYKA • 2009 • Tom 13 • Zeszyt 2

Antoni Ligęza*

AND-OR Graph with Knowledge Propagation Rules as a Model for Constraint Satisfaction Problems**

1. Introduction

A Constraint Satisfaction Problem (CSP, for short) [2, 1] is a problem where one has to assign certain variables some values. These values are restricted to belong to predefined sets – their domains. The assignment must satisfy a set of predefined constraints.

CSP is a common model for diversity of practical, theoretical, toy and entertainment problems, with Sudoku being a perfect example. In industrial practice CSP may serve as a model for planning, scheduling, resources allocation, etc. In theoretical research it is a model for logical formulae satisfaction checking (the so-called SAT problem) and mathematical programming. There are many toy problems illustrating the ideas, for example the so-called cryptoarithmetic problems, logical puzzles, or assignment problems, e.g. the Einstein (or Zebra) problem.

A specific CSP can have exactly one solution (this is the case of various puzzle and toy problems mostly), can have a large number of solutions, and can have no solution at all. The last case is the case of the over constrained problems, where no value assignment satisfying all the constraints exists.

Solving a generic CSP is both conceptually simple and intractable. It is conceptually simple since it can be accomplished – well, at least in theory – by consecutive scanning of all the elements of the Cartesian Product of all the domains. Such a Cartesian Product defines the search space for the solutions. By checking for each element all the constraints all the solutions can be found. If all the domains are finite, so is the Cartesian Product of them. Hence the algorithm for finite domains always finds all the solutions, provided that it is not an over-constrained problem. The problem is that CSP suffers from combinatorial explosion with respect to size of the search-space of potential solutions. Such problems are referred to as intractable. More precisely, intractable problems are ones for which it is known that there is no polynomial algorithm for finding the solution [2].

* Department of Automatics, AGH University of Science and Technology, Kraków and PWSZ Tarnów, Poland

** Research carried out within AGH Internal Project No. 10.10.120.105 and 11.11.120.859

Depending on the nature of the domains there are CSP formulations for Finite Domains (CSP over finite domains; Finite Domains are often shortened to FD) or continuous domains (e.g. over the rational numbers R). The finite domains can be binary ones (e.g. as in the SAT problem), composed of a finite subset of integers (e.g. digits) or symbolic ones (composed of symbols). In this paper we are focused on CSP over FD with mostly symbolic and numerical values.

The general idea behind solving CSP consist in (i) ordering the variables, (ii) subsequent selection of values from the domain and assignment of one value to each of the variables, and (iii) checking if some of the constraints are not satisfied at any stage when it is possible (as early as possible, in fact). Whenever an inconsistent assignment is found, whether partial of complete one, backtracking is enforced, and a subsequent potential solution is explored. This is backtracking which is responsible for inefficiency, but for a number of problems backtrack-free solution does not exist.

In the literature the search is organized with use of the Constraint Graph, i.e. a graph modelling the structure of the constraints [2]. In such a graph nodes represent the variables and vertices show that two variables are within the scope of the same constraint. Such a graph can be used to manage the order of variable assignment but we miss the precise knowledge about the structure of the constraints. For example, if some k variables are bound by two or more different constraints it is not visible from the constraint graph.

In this paper it is proposed to model the search space with an AND-OR graph. The principal idea is that the constraints are modelled with the AND nodes, and there are as many AND nodes as there are basic constraints constraints. All the variables involved in a constraint are OR nodes, and from any OR node there is a link to all the appropriate AND nodes. The values of the domains are represented by simple nodes pointing to the appropriate OR nodes representing the variables. For solving a CSP represented in the form of an AND-OR graph search techniques over AND-OR graphs can be applied in a direct way.

In order to make the search even more efficient the constraints can be divided into two groups: (i) ones represented as the AND nodes as such, and (ii) ones represented in the form of constructive inference rules. The explicit representation of constraints in the form of inference rules has an obvious advantage: once the values assigned to variables occurring in preconditions are known, such a rule can be fired and values of the variables occurring in the consequent part become known. They can be immediately used either in further search or – in case of inconsistency – for enforcing immediate backtracking.

The research is motivated by the so-called Consistency-Based Diagnostic Reasoning – a well-known model of a model-based diagnostic approach [4]. In this kind of diagnostic inference one tries to assign faults to system component, so that having the knowledge of system model its abnormal behavior can be fully explained. The faulty behavior of components can be of binary type, i.e. just faulty or correct, or a more extended characteristics including the qualitative or enumerative type of faults can be used.

The proposed approach is illustrated with an example of circuit diagnosis. It is a well-known example of a multiplier-adder, widely explored in both diagnostics domain [4, 3], as well as in the CSP domain [2]. It was selected here since both the problem structure and characteristics fit well into the discussed CSP modeling paradigm.

2. Constraint Satisfaction Problems. Basic Formulation

A Constraint Satisfaction Problem is one where the goal consists in finding a legal assignment of values to a finite set of predefined variables so that a set of given constraints is satisfied.

More formally, after [2], let $X = \{X_1, X_2, ..., X_n\}$ denote a set of variables, $D = \{D_1, D_2, ..., D_n\}$ is a set of domains for the variables in X and C is a set of constraints. Each constraint is given by a pair (S_i, R_i), where S_i is referred to as the scope (or scheme) and consists of a selection of variables from X while R_i is a relation defined over a Cartesian Product of domains appropriate for the variables in the scope. The relation R_i can be defined explicitly, i.e. by listing all its tuples, but more frequently it is defined implicitly by used of logical or algebraic constraints. The Constraint Satisfaction Problem (CSP) is given by the triple (X, D, C).

A solution to a CSP given by (X, D, C) is any assignment of values to variables of X of the form $\{X_1 = d_1, X_2 = d_2, ..., X_n = d_n\}$, such that $d_i \in D_i$, $i \in \{1, 2, ..., n\}$ and for any constraint $(S_i, R_i) \in C$, R_i is satisfied by the appropriate projection of the solution vector $(d_1, d_2, ..., d_n)$ over variables of S_i. Obviously, all the constraints must be satisfied, and there can be more one or many solutions; no solution may exist for an over-constrained problem.

The basic technique for solving a CSP given by (X, D, C) consists in subsequent assignment of admissible values to variables of X; the order is chosen in an arbitrary way, and it can influence how fast a solution is found.

In order to improve search efficiency both heuristics and strategies are used. The most typical strategies are based on (i) variable ordering, (ii) values ordering, and (iii) look-ahead techniques. Especially the look-ahead strategies can improve search efficiency. The principal idea is that the algorithm looks how current decisions will affect the future search.

3. AND-OR Graph Model for CSP

In this section a model for organizing the search space for CSP is put forward.

Let C denote the set of constraints, $C = \{C_1, C_2, ..., C_k\}$, and $C_i = (S_i, R_i)$, where S_i is the constraint scheme, and R_i is a relation defining the constraint; $i \in \{1, 2, ..., k\}$. Every $C_i \in C$ is assigned to a single AND-node of the AND-OR graph. The AND nodes form the top-level of the AND-OR graph.

The OR nodes are assigned the variables of X. There is a link from any OR-node corresponding to variable X_j to the node representing constraint C_i if and only if variable X_i belongs to scheme S_i of C_i. The OR nodes form the medium level of the AND-OR graph.

Let $U = D_1 \cup D_2 \cup \dots D_n$ denote the union (the algebraic union of all the domains). The lowest level of the graph is formed by a set of nodes corresponding to the elements of U. There is a link from $d \in U$ to an OR node above if and only if this element belongs to the domain of the variable assigned to this OR node.

An example AND-OR graph is presented in Figure 1. In the figure there are two variables, namely $V1$ and $V2$ constituting the OR-nodes. They must satisfy a single constraint $C1$ assigned to the topmost AND-node. The definition of this constraint is simple: both $V1$ and $V2$ must take a single value from the appropriate domain. The domain of $V1$ is $D1 = \{m1, m2, a1\}$, and the domain of $V2$ is $D2 = \{m1, m3, a1, a2\}$. As it can be noticed, these domains are overlapping. The elements of them form the lowest level presenting admissible values of the variables (Fig. 1).

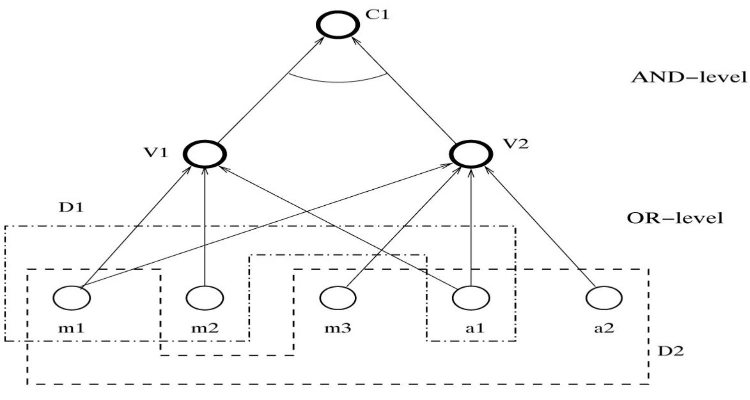

Fig. 1. An example AND/OR graph for a single-constraint two variables CSP

The AND-OR graph presented in Figure 1 will be used further on when analyzing an example diagnostic problem.

4. Rules for Constraint Propagation

In this section we introduce the concept of rules for constraint propagation. Such rules can be used to refine the search for solution and hence improve search efficiency.

Let us refer to the basic model of CSP, as presented in section 2. There is only one set X of variables which are the decision variables, i.e. ones to which certain values must be assigned according to the decision of the end-user so that all the constraints are satisfied. For a large class of problems, however, it seems reasonable to introduce an independent set of variables which can just be observed. Their values depend on the state of the system and perhaps on the current assignment of the values to variables of X. We shall call the set M and refer to values of these variables as observations or manifestations. For intuition, the role of these variables is that a specific observation can confirm or reject some variable assignment hypotheses.

Let $\bar{Y}$ be a sequence of variables and let $\bar{d}$ denotes a sequence of values of the same length. By $\bar{Y} = \bar{d}$ we shall denote a conjunctive assignment of values of $\bar{d}$ to variables of $\bar{Y}$. A general form of the proposed constraint propagation rules is

$$\bar{X}_L = \bar{d}_L \wedge \bar{M}_L = \bar{m}_L \rightarrow \bar{X}_R = \bar{d}_R \wedge \bar{M}_R = \bar{m}_R$$

where $\bar{X}_L$ and $\bar{X}_R$ are sequences of some different variables from *X*, and $\bar{M}_L \bar{M}_R$ are sequences of some different variables from *M*.

Rules as above can take the presented above and eight different subforms, i.e. the part of variables from *X* or *M* can be missing in the preconditions, or conclusions or both in the preconditions and conclusions.

The application of such rules can consists in:

- constraint propagation in the form of explicit restriction of some variables of *X* from the conclusion part,
- consistency checking, i.e. confirming or rejecting variable assignment defined in the precondition part through comparison of the variable values of the *M* part with the current observations.

In the next section we introduce an example from the area of automated diagnosis and next we show application of the AND-OR graphs with rules for refinement of potential diagnoses.

5. Automated Diagnosis. An Example

Automated diagnosis is an activity oriented towards the detection of faulty behavior and its explanation, i.e. isolation of faulty components responsible for the observed misbehavior of the analyzed system. Consistency-based diagnosis [4] is performed by detecting and analyzing inconsistency between system model (or, more precisely, its selected outputs) and current observations. Diagnostic reasoning leads to isolation of faulty components responsible for the observed misbehavior of the analyzed system. In order to do that, sets of elements which cannot all work correctly (the so-called conflict sets or conflicts) are identified first. Potential diagnoses are generated as certain kind of intersection of the conflict sets [4].

Let us consider the circuit presented in Figure 2, consisting of three multipliers, *m*1, *m*2, and *m*3, and two adders, *a*1 and *a*2 [4]. The inputs are $A = 3$, $B = 2$, $C = 2$, $D = 3$, and $E = 3$. Consider the example inconsistency "*F* is observed to be 10, not 12".

The prediction that $F = 12$ depends on the correct operation of *a*1, *m*1, and *m*2. Since *F* is not 12, at least one element of *a*1, *m*1, and *m*2 is faulty. Thus the set {*a*1, *m*1, *m*2} is a conflict responsible the misbehaving variable *F*.

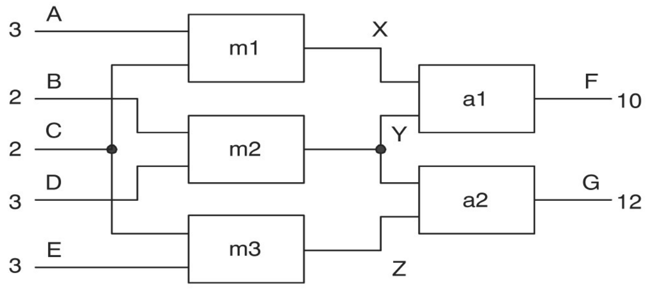

Fig. 2. The multiplier-adder example circuit

The second way to calculating *F* is as follows: from *A* and *C* through *m*1 we calculate *X* = 6, from *C*, *E* through *m*3 we have Z = 6, from *Z* and *G* through *a*2 we expect that *Y* = 6, and finally from *X*, and *Y* through *a*1 we calculate *F* = 12. This prediction that *F* = 12 depends on the correct operation of *a*1, *a*2, *m*1 and *m*3. Thus we have second conflict {*a*1, *a*2, *m*1, *m*3} for misbehaving variable *F*.

Using these conflict sets ({*a*1, *m*1, *m*2}, {*a*1, *a*2, *m*1, *m*3}) we can now generate minimal diagnoses as minimal hitting sets, i.e. sets having exactly one element from each conflict. The diagnoses are as follow: {*a*1}, {*m*1}, {*m*2, *m*3}, {*a*2, *m*2}. These diagnoses are modelled by solution subgraphs of the AND-OR graph in Figure 1.

This example is discussed in more details in [3].

6. AND-OR Graph With Rules: Multi-Mode Diagnoses

A more general constraint satisfaction framework for multi-element multi-mode diagnosis can be stated in the following way. Let *V* denote a set of variables, $V = X \cup M$. *M* is the set of observable (measurable) system variables, and *X* is the set of diagnostic variables aimed at describing different faulty modes of diagnosed components. By nature these sets are pairwise mutually disjoint.

In the case of the example system we have: *M* = {*A*, *B*, *C*, *D*, *E*, *F*, *G*, *X*, *Y*, *Z*} and the set of decision variables is *X* = {*m*1, *m*2, *m*3, *a*1, *a*2}. Variables of *M* take the values of the real numbers (or integers) restricted to some reasonable intervals, while variables of *X* are restricted to some of the possible modes of misbehavior. In our case the values are restricted to {–, 0, +} with the obvious meaning of lowering the output value, producing correct output and producing the output higher than expected (the so-called reference value). For example, *m*1(+) means that *m*1 being faulty produces output higher than expected by the reference value.

Finally, let *R* be a set of rules defining all the accessible knowledge about wrong behavior of faulty components depending on the faulty mode; examples of such rules are shown below. The analysis of potential qualitative diagnoses is performed by propagation of values of variables of M. Whenever an inconsistency of the expected and observed values is encountered, the candidate diagnosis is rejected.

Let us analyze in turn all the four potential diagnoses and their potential qualitative forms. The analysis is aimed at finding all admissible qualitative diagnoses.

Case of $m1$: There are two modes of faulty operation of $m1$, i.e. $m1(-)$ and $m1(+)$.

Consider $m1(-)$ first. An obvious rule is $A(0) \wedge B(0) \wedge m1(-) \rightarrow X(-)$. The reference value for X (if $m1$ is correct) is 6. In case of $m1(-)$ we must have $X(-)$ denoting in fact a value lower than 6. Since $a1$ is correct, with rule $X(-) \wedge Y(0) \wedge a1(0) \rightarrow F(-)$ we conclude $F(-)$. This is consistent with observations since $F = 10$, and the reference value is 12.

Now, consider $m1(+)$. An obvious rule is $A(0) \wedge B(0) \wedge m1(+) \rightarrow X(+)$. The reference value for X (if $m1$ is correct) is again 6. In case of $m1(+)$ we should have $X(+)$ denoting some value greater than 6. Since $a1$ is correct, with rule $X(+) \wedge Y(0) \wedge a1(0) \rightarrow F(+)$ we conclude $F(+)$. This is inconsistent with observations since $F = 10$, and the reference value is 12. Hence the diagnosis $m1(+)$ is inconsistent with observations and, as such, rejected.

Case of $a1$: There are two modes of faulty operation of $a1$, i.e. $a1(-)$ and $a1(+)$.

Consider $a1(-)$ first. An obvious rule is $X(0) \wedge Y(0) \wedge a1(-) \rightarrow F(-)$. The reference value for F (if $a1$ is correct) is 12. In case of $a1(-)$ we have $F(-)$ denoting in fact some value lower than 12. This is consistent with observations since $F = 10$, which is lower than 12.

On the other hand consider $a1(+)$. An obvious rule is $A(0) \wedge B(0) \wedge a1(+) \rightarrow F(+)$. The reference value for F (if $a1$ is correct) is again 12. In case of $a1(+)$ we would have $F(+)$ denoting value greater than 12. This is inconsistent with observations since $F = 10$, and the reference value is higher. Hence the diagnosis $a1(+)$ is inconsistent, and in consequence – rejected.

Case of $\{a2, m2\}$: There are four combined potential faulty modes, i.e. $\{a2(-), m2(-)\}$, $\{a2(-), m2(+)\}$, $\{a2(+), m2(-)\}$, $\{a2(+), m2(+)\}$. Let us analyze them in turn.

Case $\{a2(-), m2(-)\}$: The reference value for Y (if $m2$ is correct) is 6. With rule $B(0) \wedge D(0) \wedge m2(-) \rightarrow Y(-)$ we should have $Y(-)$ denoting in fact value lower than 6. Since $a1$ is correct, this implies also $F(-)$.

This is consistent with observations since $F = 10$, and the reference value is 12. On the other hand, we have Y (–) at the input of a2, and with the obvious rule $Y(-) \wedge Z(0) \wedge a2(-) \rightarrow G(-)$ we should have $G(-)$. Simultaneously, the reference value for G is 12 and it is equal to the observed value. Hence diagnosis {a2(–), $m2(-)$} must be rejected as an inconsistent one.

Case $\{a2(-), m2(+)\}$: The reference value for Y (if $m2$ is correct) is 6. With rule $B(0) \wedge D(0) \wedge m2(+) \rightarrow Y(+)$ we should have $Y(+)$ denoting in fact a value greater than 6. Since a1 is correct, with rule $X(0) \wedge Y(+) \wedge m1(0) \rightarrow F(+)$ we should have $F(+)$, i.e. a value greater than 12. This is inconsistent with observations since $F = 10$, and the reference value is 12. Hence diagnosis $\{a(-), m2(+)\}$ must also be rejected as an inconsistent one.

Case $\{a2(+), m2(-)\}$: The reference value for Y (if $m2$ is correct) is 6. In case of $m2(-)$ with rule $B(0) \wedge D(0) \wedge m2(-) \rightarrow Y(-)$ we have $Y(-)$ denoting in fact value lower than 6.

Since a1 is correct, this implies also $F(–)$. This is consistent with observations since $F = 10$, and the reference value is 12. On the other hand, we have $Y(–)$ at the input of $a2$. A rule of the form $Y(–) \wedge Z(0) \wedge a2(+) \rightarrow G(?)$ is not a decisive one. In fact $G(?)$ may cover the observed value. Hence diagnosis $\{a2(+), m2(–)\}$ may be considered a consistent one.

Case $\{a2(+), m2(+)\}$: The reference value for Y (if $m2$ is correct) is 6. With rule $B(0) \wedge D(0) \wedge m2(+) \rightarrow Y(+)$ we should have $Y(+)$ denoting in fact a value greater than 6. Since a1 is correct, this implies also $F(+)$ through the rule $X(0) \wedge Y(+) \wedge a1(0) \rightarrow F(+)$. This is inconsistent with observations since $F = 10$, and the reference value is 12. Hence diagnosis $\{a2(+), m2(+)\}$ must also be rejected as an inconsistent one.

Case of $\{m2, m3\}$: There are four potential potential faulty modes, i.e. $\{m2(–), m3(–)\}$, $\{m2(–), m3(+)\}$, $\{m2(+), m3(–)\}$, $\{m2(+), m3(+)\}$. Let us analyze them in turn.

Case $\{m2(–), m3(–)\}$. The reference value for Y (if $m2$ is correct) is 6. In case of $m2(–)$ with rule $B(0) \wedge D(0) \wedge m2(–) \rightarrow Y(–)$ we should have $Y(–)$ denoting in fact value lower than 6. Since a1 is correct, this implies also $F(–)$. This is consistent with observations since $F = 10$, and the reference value is 12. On the other hand, in case of $m3(–)$ with rule $C(0) \wedge E(0) \wedge m3(–) \rightarrow Z(–)$ we also should have $Z(–)$, where the reference value is 6. Further, at both inputs of $a2$ we have values lower then reference values. By rule $Y(–) \wedge Z(–) \wedge a2(0) \rightarrow G(–)$ we should have $G(–)$. But the reference value for G is 12 and it is equal to the observed value. Hence diagnosis $\{m2(–), m3(–)\}$ must be rejected as an inconsistent one.

Case $\{m2(–), m3(+)\}$. As above, $m2(–)$ is consistent with observed $F(–)$. Further, in case of $m3(+)$ we have $Z(+)$ by the rule $C(0) \wedge E(0) \wedge m3(+) \rightarrow Z(+)$, where the reference value for Z is 6. Hence, at the inputs of $a2$ we have one lower and one higher then reference values. The rules $Y(–) \wedge Z(+) \wedge a2(0) \rightarrow G(?)$ is indecisive. We obtain $G(?)$ while the reference value for G is 12 and it is equal to the observed value.

Hence diagnosis $\{m2(–), m3(+)\}$ may be considered as a consistent one.

Case $\{m2(+), m3(–)\}$. The reference value for Y (if $m2$ is correct) is 6. With rule $B(0) \wedge D(0) \wedge m2(+) \rightarrow Y(+)$ we should have $Y(+)$ denoting in fact a value greater than 6. Since a1 is correct, this implies also $F(+)$ through the rule $X(0) \wedge Y(+) \wedge a1(0) \rightarrow F(+)$. This is inconsistent with observations since $F = 10$, and the reference value is 12. Hence diagnosis $\{m2(+), m3(–)\}$ must also be rejected as an inconsistent one.

Case $\{m2(+), m3(+)\}$. As above, with rule $B(0) \wedge D(0) \wedge m2(+) \rightarrow Y(+)$ and assuming that a1 is correct we have $F(+)$ through the rule $X(0) \wedge Y(+) \wedge a1(0) \rightarrow F(+)$. This is inconsistent with observations since $F = 10$, and the reference value is 12. Hence diagnosis $\{m2(+), m3(+)\}$ must also be rejected as an inconsistent one.

7. Concluding Remarks

The paper presents an AND-OR graph model for constraint satisfaction problems. The proposed model is completed with auxiliary inference rules for propagation of variable values. This simple approach can be applied for efficient elimination of inconsistent variable assignments. As a consequence, the set of potential solutions can be further refined. An example of application to qualitative diagnosis with respect to deviation sign for faulty components is presented.

References

[1] Apt. K.R., *Principles of Constraint Programming.* Cambridge University Press, Cambridge, UK.
[2] Dechter R., *Constraint Processing.* Morgan Kaufmann Publishers, San Francisco, CA, 2003.
[3] Ligęza A., Kościelny J.M., *A new approach to multiple fault diagnosis. combination of diagnostic matrices, graphs, algebraic and rule-based models. The case of two-layer models.* Int. J. Appl. Math. Comput. Sci., 18(4), 2008, 465–476.
[4] Reiter R., *A theory of diagnosis from first principles*. Artificial Intelligence, 32, 1987, 57–95.