Ordering Problem Subgoals

Jie Cheng and Keki B. Irani
Artificial Intelligence Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109-2122, USA

Abstract

Most past research work on problem subgoal ordering are of a heuristic nature and very little attempt has been made to reveal the inherent relationship between subgoal ordering constraints and problem operator schemata. As a result, subgoal ordering strategies which have been developed tend to be either overly committed, imposing ordering on subgoals subjectively or randomly, or overly restricted, ordering subgoals only after a violation of ordering constraints becomes explicit during the development of a problem solution or plan. This paper proposes a new approach characterized by a formal representation of subgoal ordering constraints which makes explicit the relationship between the constraints and the problem operator schemata. Following this approach, it becomes straightforward to categorize various types of subgoal ordering constraints, to manipulate or extend the relational representation of the constraints, to systematically detect important subgoal ordering constraints from problem specifications, and to apply the detected constraints to multiple problem instances.

1 Introduction

Subgoal ordering plays such an important role in planning and problem solving that a great amount of research has been dedicated to detecting subgoal ordering constraints and applying the constraints to problem space search control [Chapman, 1987, Dawson and Siklossy, 1977, Ernst and Goldstein, 1982, Sacerdoti, 1974, Sacerdoti, 1975, Sacerdoti, 1977, Tate, 1975, Waldinger, 1981, Warren, 1974]. However, most of the reported approaches are heuristic and the subgoal ordering constraints are not well-defined. Further, very little attempt has been made to reveal the inherent relationship between subgoal ordering constraints and problem operator schemata. Chapman [1987] was the first who gave a formal account to the subgoal ordering problem, but he has not addressed the relationship between ordering constraints and problem operator schemata. Ernst and Goldstein [1982] tried to elucidate the relationship between ordering constraints and problem operators, but their approach requires the use of instances of problem operator schemata and cannot guarantee the correctness of the generated ordering of subgoals in general (see [Irani and Cheng, 1987]). Consequently, subgoal ordering strategies previously developed tend to be either overly committed, imposing ordering on subgoals subjectively or randomly, or overly restricted, ordering subgoals only after a violation of ordering constraints becomes explicit during the development of a problem solution or plan.

In our research, an approach is developed to explicitly represent subgoal ordering constraints. Based on this representation, procedures are then constructed to systematically detect the constraints. This approach makes it possible to detect ordering constraints without getting involved in planning or problem solving.

Our approach proves to be advantageous in that once a representation for a class of constraints is constructed, its properties can be studied and the representation can be manipulated to extend its generality or to produce formulations for new types of constraints. Another advantage of our approach over the old ones is that much time formerly devoted to detecting violations of constraints and ordering/reordering partial problem solutions or plans can now be saved. The constraints can be derived from problem specifications via reasoning and henceforth, problem subgoals can be properly ordered even before problem solving or planning. Finally, constraints among a group of problem subgoals, once derived, can be stored and applied to multiple problems as long as they share the same problem operators and involve at least these subgoals. The complexity of the approach is measured to be polynomial with respect to the number of subgoals involved, with the assumption that in any rule schema, all that is implied by the preconditions or the postconditions is explicitly represented.

In [Irani and Cheng, 1987], our initial results in subgoal ordering have been reported. In this paper, we present an extension to our previous work. The paper is organized as follows. First, an extended representation schema for subgoal ordering constraints is defined. The features of the represented constraints are then discussed. A method to reduce the complexity of constraint detection is described and then several procedures for detecting such constraints are presented. Finally, an example is used to further illustrate this approach.

2 Representing Subgoal Ordering Constraints

In this section, basic notations used in the paper are introduced, the initial results of our research reported in [Irani and Cheng, 1987] are briefly reviewed, and then an extended schema for representing subgoal ordering constraints is presented.

2.1 Notations

To represent subgoal ordering constraints, three components of a problem model are needed, namely, a state space, a set of problem operators and a goal specification. It is assumed that each state is represented by a conjunction of propositions. *G* is a conjunction of literals each of which is a subgoal. With this restriction, the goal *G* can be equivalently represented as a set of subgoal literals.

In this paper, *s* is a symbol representing a state. *T* (with a suitable subscript) denotes a problem operator which transforms one state into another, *preck* and *postk* are the precondition formula and postcondition formula respectively of operator T_k. S_f stands for a subset of states in which predicate formula / holds true. *g* (with a suitable subscript) denotes a subgoal. A problem solution is a sequence of states, $(S_1\ S_2,\ S_n)$, where s_l is an initial state and s_n is the first state that satisfies the goal condition. We say "a subgoal *g* is achieved at the *m-th* step of a solution *(s\,S2, ..,*s_n*)* if s, € S_g for $^m < i <^n$ and if m > 1 then s_m_i € S_g. *g* is said to be trivially achieved in a solution if *m* = 1. We say "a subgoal *gi* precedes a subgoal *gj* in a solution *(si*, S2, ••-, S*n)"* if *gi(gj)* is achieved at the *rni(rri2)-th* step of the solution and *mi* < ^{m}2-

2.2 Review

In our previous paper [Irani and Cheng, 1987] , a binary relation <* over a set of subgoals *G* was defined. This relation was then proved to be a formal characterization of a type of strong ordering constraints among the subgoals, namely, the constraint that *a subgoal gi must precede another subgoal gj in all problem solutions in which both art non-trivially achieved.* In later discussion, this type of constraints will be referred to as the constraints of type <*.

As demonstrated in our earlier work, the relational representation of problem subgoal ordering constraints facilitated the development of procedures for systematically detecting the constraints. Furthermore, the representation reveals that there exists a generic relationship between subgoal ordering constraints and a problem operator schema. Therefore, it is possible to detect a class of ordering constraints among subgoals without getting involved in problem solving or planning processes.

The relation <*, however, has two weaknesses. First, the relation is binary, so only pairs of subgoals are examined for the detection of ordering constraints. In this way, the relation cannot be used to identify a constraint among two subgoals if that constraint is dependent on the coexistence of other subgoals. For example, let three subgoals be defined for a robot planning problem. The first subgoal is for the robot to place a block, b1, next to another block, b2. The second subgoal is for b1 to be in *roorni* and the third is for the robot to be in another room, say, *roorri2.* It is obvious that the robot has to achieve *Nextto(*1,62) before it achieves *Inroom(robot,r001712).* However, < *Nextto(b*, 62), *Inroom(robot, roorri2)* > is not in the relation <*, because in this case, the ordering constraints can be identified only when all the three subgoals are taken into consideration.

Another weakness of the relation <* is that in order to detect ordering constraints ahead of planning or problem solving, every pair of subgoals has to be screened. This is not very economical because constraints often exist among only a few problem subgoals. These weaknesses of the previous approach motivated us to extend the representation of subgoal ordering constraints to circumvent the limitations while preserving the declarative feature and the simplicity of relation <*.

2.3 Extension

The extension to the constraint representation is basically to let more subgoals be considered simultaneously for the detection of constraints. This extension should not only make it possible to detect constraints among pairs of subgoals that are dependent on the coexistence of other subgoals, but also pave the way for an efficient top-down approach in detecting subgoal ordering constraints. In the following, we define a relation -<$^+$. Here, {gk,} and *{gh }* represent sets of arbitrary subgoals and Atpjb, represents the conjunction of the elements of { G K } .

Definition 1 *(Extension 1)* $\prec^+$ *is a binary relation over* G', *where* $G' = 2^G$. *For any* $\{g_{k_i}\} \in G'$ *and any* $\{g_{h_j}\} \in G'$, $\{g_{k_i}\} \prec^+ \{g_{h_j}\}$ *iff*

$$\forall l \forall s (T_l(s) \in S_{\wedge_{i,j} g_{k_i}, g_{h_j}} \longrightarrow s \in S_{\wedge_i g_{k_i}})$$
$$\bigwedge \forall g \in \{g_{h_j}\} \exists l \exists s (T_l(s) \in S_{\wedge_i g_{k_i}, g_{h_j}} \wedge s \notin S_g)$$

This relation can be verbally stated as follows: $\{g_{k_i}\} \prec^+ \{g_{h_j}\}$ if and only if, for every rule, say T_l, if the rule can transform a state s into a state $T_l(s)$ satisfying all subgoals of $\{g_{k_i}\} \cup \{g_{h_j}\}$, then $\wedge_i g_{k_i}$ must have already been satisfied in s, whereas for each subgoal g in $\{g_{h_j}\}$, there exists at least one rule and one state s such that g is not satisfied in the state s. For example, if we take the problem goal to be the conjunction of the three subgoals we mentioned before, and we use the same set of problem operators as used by Sacerdoti [1974], then we can derive relation $\prec^+$ to be like:

$$\begin{aligned}\prec^+ = \{ &< Inroom(b_1, room_1), Nextto(b_1, b_2) >,\\ &< Inroom(b_1, room_1), Inroom(Robot, room_2) >,\\ &< Inroom(b_1, room_1) \wedge Nextto(b_1, b_2),\\ &\quad Inroom(Robot, room_2) > \}\end{aligned}$$

Obviously, the constraint that was missed by $<^*$ is now identified by $\prec^+$, namely, g_1 must be achieved before g_3, given the coexistence of g_2. The type of constraints characterized by $\prec^+$ is given by theorem 1 and will be referred to as the constraints of type $\prec^+$.

Theorem 1 [1] $\{g_{k_i}\} \prec^+ \{g_{h_j}\}$ *iff (1) each $g \in \{g_{k_i}\}$ precedes the complete accomplishment of $\wedge_j g_{h_j}$ for any problem solution in which $\wedge_i g_{k_i}$ and $\wedge_j g_{h_j}$ are non-trivially achieved,*

(2) for each $g \in \{g_{h_j}\}$, there exists at least one problem solution in which all subgoals of $\{g_{k_i}\} \cup \{g_{h_j}\}$ are non-trivially achieved and in which g does not precede the complete accomplishment of the subgoals in $\{g_{h_j}\} - \{g\}$.

The relation $\prec^+$ has three desirable properties as described in the following:

Property 1 *If $\{g_{k_i}\} \prec^+ \{g_{h_j}\}$, then the relation $\prec^+$ restricted to $2^{\{g_{h_j}\}}$ is empty.*

The implication of this property is that, once we derive a constraint of type $\prec^+$ among two sets of subgoals, namely, $\{g_{k_i}\} \prec^+ \{g_{h_j}\}$, we do not need to go further to search for the constraints of type $\prec^+$ among the subgoals in $\{g_{h_j}\}$.

Property 2 *If $\{g_{k_i}\} \prec^+ \{g_{h_j}\}$ and $\{g'_{k_i}\} \prec^+ \{g'_{h_j}\}$, and $\{g_{k_i}\} \cup \{g_{h_j}\} = \{g'_{k_i}\} \cup \{g'_{h_j}\}$ then $\{g_{k_i}\} = \{g'_{k_i}\}$ and $\{g_{h_j}\} = \{g'_{h_j}\}$.*

Property 2 is the uniqueness property of the relation $\prec^+$. It is equivalent to saying that there is at most one partition of a set of subgoals into two blocks such that there is a constraint of type $\prec^+$ between the two blocks of the subgoals.

Property 3 *If for any partition $\{G_1, G_2\}$ of a set G of subgoals, $G_1 \not\prec^+ G_2$ and $G_2 \not\prec^+ G_1$, then $\prec^+$ restricted to G is empty.*

The three properties of $\prec^+$ can be used to greatly expedite the process of detecting subgoal ordering constraints. Specifically, a top-down approach can be used which starts by detecting constraints among a set of subgoals and then, only if a constraint is found, proceeds to detect constraints among the subsets of subgoals. The rest of the extension work makes the constraint detection even more efficient.

It is clear from the above discussion that $\prec^+$ is much more useful than relation $<^*$. However, $\prec^+$ still has one disadvantage. As can be seen from theorem 1, $\prec^+$ identifies much more than we actually need. What we need is the precedence relation between pairs of subgoals that are nontrivially achieved. Therefore, it will be a waste to detect the constraints we need via the derivation of relation $\prec^+$. This problem leads to the introduction of a new relation as follows:

Definition 2 *(Extension 2:) $\prec^x$ is a binary relation from 2^G to G. For any $\{g_{k_i}\} \in 2^G$ and any $g \in G$,* $\{g_{k_i}\} \prec^x g$ iff

$$\forall l \forall s (T_l(s) \in S_{\wedge_i g_{k_i} \wedge g} \longrightarrow s \in S_{\wedge_i g_{k_i}})$$
$$\bigwedge \exists l \exists s (T_l(s) \in S_{\wedge_i g_{k_i} \wedge g} \wedge s \notin S_g)$$

Actually, $\prec^x$ is a sub-relation of $\prec^+$, because $\prec^x = \prec^+ \cap 2^G \times G$. The link between the relation $\prec^x$ and the constraint it identifies is as follows:

[1] Proofs of theorems have not been given here due to space restriction. Interested readers are referred to [Cheng, 1989].

Let $\{g_{k_i}\} \cup \{g\} \subseteq G$. $\{g_{k_i}\} \prec^x g$ if and only if for each $g' \in \{g_{k_i}\}$, g' precedes g in any problem solution where all subgoals of $\{g_{k_i}\} \cup \{g\}$ are non-trivially achieved.

The improvement of $\prec^x$ over $\prec^+$ is that $\prec^x$ can directly identify ordering constraints among pairs of subgoals that are dependent on the existence of other subgoals. However, as indicated by its three properties, $\prec^+$ has the potential of speeding up the the process of constraint detection. Therefore, both relations are employed in our development of constraint detecting procedures.

3 Detecting Subgoal Ordering Constraints

Now that a type of subgoal ordering constraints is explicitly represented, we can proceed to build procedures to systematically detect the constraints from problem specifications. In this section, we first present procedures for detecting constraints of type $\prec^x$ among a set of subgoals, and then, a procedure for detecting constraints of type $\prec^+$. To reduce the complexity of detecting the constraints, we first introduce a notion called "the most effective subrelation of $\prec^x$" which is a special subset of $\prec^x$.

3.1 $\prec^x_{min}$: A Special Subrelation of $\prec^x$

The constraints that concern us are of the following type: *a subgoal g_i must precede another subgoal g_j in any solution in which a set of subgoals including g_i and g_j are nontrivially achieved.* The relation $\prec^x$ is used to detect this type of constraints. For example, if $\{g_1, g_2\} \prec^x g_3$, then g_1 will precede g_3 and g_2 will precede g_3 in any solution in which g_1, g_2 *and* g_3 are nontrivially achieved. Note that $\{g_1, g_2\} \prec^x g_3$ will be true for any problem with goal G such that $\{g_1, g_2, g_3\} \subseteq G$, as long as all other parts of the problem specification are the same. The interesting point is that, as will be shown, there always exists a minimal subset of $\prec^x$ which can identify all the constraints that are identified by $\prec^x$. Such a subset is called the **most effective subrelation of** $\prec^x$. In this section, this subset is defined and its properties described. First, we define a set C_G which contains the constraints that we are interested in.

Definition 3 *C_G is a binary relation over G. For $g_i \in G$ and $g_j \in G$, $< g_i, g_j > \in C_G$ iff there exist $g_{k_1}, g_{k_2}, \ldots, g_{k_m}$, such that $g_i \in \{g_{k_i}\}$, $g_j \notin \{g_{k_i}\}$ and $\{g_{k_i}\} \prec^x g_j$.*

Next, we define a mapping F which relates the C_G with the relation $\prec^x$.

Definition 4 *$F : 2^{\prec^x} \longrightarrow 2^{C_G}$ is an injective function. $F(\{< \wedge_i g_{k_i}, g >\}) = \cup_i \{< g_{k_i}, g >\}$ and for any $S \subseteq \prec^x$, $F(S) = \cup_{e \in S} F(\{e\})$.*

With these definitions, it is now possible to compare the relative effectiveness of any two subsets of $\prec^x$ in their identification of subgoal ordering constraints.

Definition 5 *Let $\prec^x_{sub1}$ and $\prec^x_{sub2}$ be subsets of $\prec^x$. $\prec^x_{sub1}$ is* **more effective than** *$\prec^x_{sub2}$, denoted by $\prec^x_{sub1} \sqsubset \prec^x_{sub2}$, if $|\prec^x_{sub1}| < |\prec^x_{sub2}|$*

and $F(\prec^x_{sub1}) \supseteq F(\prec^x_{sub2})$. *A subset of* $\prec^x$ *is the* **most effective subrelation of** $\prec^x$, *denoted by* $\prec^x_{min}$, *if for any* $\prec^x_{sub} \subseteq \prec^x$, $\prec^x_{min} \sqsubset \prec^x_{sub}$.

We state without proof the following theorem:

Theorem 2 *The relation* $\prec^x_{min}$ *exists and is unique.*

We are only interested in the relation C_G as defined in definition 3. The following theorem shows that the relation CG can be obtained from $\prec^x_{min}$.

Theorem 3 $F(\prec^x_{min}) = C_G$.

Taking as an example the simple robot planning problem we used in section 2.2, we can derive the following:

$$\begin{aligned} C_G = \{ & < Inroom(b_1, room_1), Nextto(b_1, b_2) >, \\ & < Inroom(b_1, room_1), Inroom(Robot, room_2) >, \\ & < Nextto(b_1, b_2), Inroom(Robot, room_2) > \} \end{aligned}$$

and

$$\begin{aligned} \prec^x_{min} = \{ \quad & < Inroom(b_1, room_1) \wedge Nextto(b_1, b_2), \\ & Inroom(Robot, room_2) >, \\ & < Inroom(b_1, room_1), Nextto(b_1, b_2) > \} \end{aligned}$$

3.2 *GENCON* and *GENCON2*

In this section, two recursive procedures called *GENCON* and *GENCON2* are described. *GENCON* is the main procedure which generates $\prec^x_{min}$ for a given goal *G*. *GENCON2* is called by *GENCON*. Another procedure, *ONECON*, which is a called by both *GENCON* and *GENCON2*, detects a $\prec^+$ relation among a set of subgoals, if any. *ONECON* is described in the next section. The parameters *GS* and R_{min} for *GENCON* are initially set to *G* and ϕ respectively.

```
Procedure: $GENCON(GS, R_{min})$
Input: $GS$ — a set of subgoals;
Output: $R_{min} = \prec^x_{min}$.
VAR: $C$ — holds $ONECON$'s return value which is either a member of $\prec^+$ or $\phi$.
  1. If $|GS| = 1$, then RETURN;
  2. Call $ONECON(GS, C)$;
  3. If $C = \phi$, namely, there exists no $\prec^+$ relationship
     among the subgoals of $GS$, then RETURN;
  4. If $C = \{g_{k_i}\} \prec^+ \{g\}$ (equivalently, $\{g_{k_i}\} \prec^x g$), where
     $\{g_{k_i}\} \in 2^{GS}$, $g \in GS$ and $\{g_{k_i}\} \cup \{g\} = GS$, then
     $R_{min} = R_{min} \cup \{< \{g_{k_i}\}, g >\}$;
  5. If $C = \{g_{k_i}\} \prec^+ \{g_{h_j}\}$, where $\{g_{k_i}\} \in 2^{GS}$, $\{g_{h_j}\} \in$
     $2^{GS}$, $\{g_{k_i}\} \cup \{g_{h_j}\} = GS$, and $|\{g_{h_j}\}| > 1$, then for
     all $g \in \{g_{h_j}\}$ call $GENCON2(\{g_{k_i}\}, g, R_{min})$;
  6. call $GENCON(\{g_{k_i}\}, R_{min})$;
  7. RETURN
```

GENCON2 is a procedure called by *GENCON*. It is used to generate all the $\prec^x$ constraints among subgoals of $GS \cup \{g\}$ with the knowledge that *g* cannot precede any of the subgoals in *GS*.

```
Procedure: $GENCON2(GS, g, R_{min})$
Input: $GS$ — a set of subgoals; $g$ — a single subgoal;
Output: $R_{min} \subseteq \prec^x_{min}$.
VAR: $C$ — holds $ONECON$'s return value which is either a member of $\prec^+$ or $\phi$.
  1. Call $ONECON(GS \cup \{g\}, C)$;
  2. If $C = \phi$, namely, there exists no $\prec^+$ type con-
     straints among the subgoals of $GS \cup \{g\}$, then RE-
     TURN;
  3. If $C = \{g_{k_i}\} \prec^+ \{g\}$ (equivalently, $\{g_{k_i}\} \prec^x g$), where
     $\{g_{k_i}\} = GS$, then $R_{min} = R_{min} \cup \{< \{g_{k_i}\}, g >\}$;
  4. If $C = \{g_{k_i}\} \prec^+ \{g_{h_j}\}$, where $\{g_{k_i}\} \in 2^{GS}$, $\{g_{h_j}\} \in$
     $2^{GS \cup \{g\}}$, $\{g_{k_i}\} \cup \{g_{h_j}\} = GS \cup \{g\}$, then call
     $GENCON2(\{g_{k_i}\}, g, R_{min})$;
  5. RETURN
```

The two procedures listed above are developed following the guidelines provided by the properties of the relation $\prec^+$. For example, in *GENCON*, step 3 follows from property 3; step 4 follows from property 2; step 5 and step 6 follow from property 1. It has been proved that, as desired, the set of elements generated by the procedure *GENCON* is exactly $\prec^x_{min}$.

3.3 *ONECON*

A basic component of the procedures *GENCON* and *GENCON2* is a procedure, named *ONECON*, which identifies a member of $\prec^+$, or a constraint of type $\prec^+$, for a set of subgoals. This section describes the development of *ONECON*. A new membership criterion for the relation $\prec^+$ is constructed. The reason is that, although the original definition of $\prec^+$ provides a simple characterization of constraints, it cannot be directly implemented in detecting the constraints because that would necessitate exhaustive search of the whole problem space. The new membership criterion provides an operational definition for $\prec^+$ which is almost directly translated into the procedure *ONECON*.

We use Strips-like problem representation for the following discussion, although our procedure can be applied to other representations as well. $post_k$ will henceforth be used to denote the formulas in the *Add-list* of rule T_k.

Definition 6 *A binding* π *of* $post_k$ *with a goal G is denoted by* $post_k^\pi$ *and is the result of assigning constants in G to corresponding variables of matching formulas in* $post_k$ *and assigning arbitrary constants to the rest of the variables. A binding* π *of* $prec_k$ *with a goal G is denoted by* $prec_k^\pi$ *and is the result of passing the binding from* $post_k$ *to* $prec_k$ *and assigning arbitrary constants to the remaining variables.*

Definition 7 *A goal G is consistent with a post-condition* $post_k$ *under binding* π*, or simply,* $con(post_k^\pi, G)$*, if (1) for all* $g \in G$*, it is not true that* $\neg g$ *is a logical consequence of* $post_k^\pi$ *and the preserved* $prec_k^\pi$ *formulas*[2]*;*

[2]Preserved $prec_k^\pi$ formulas are the predicates in $prec_k^\pi$ that are not deleted by the rule T_k.

(2) None of the problem constraints[3] is violated by the conjunction of $post_k^\pi$, G, and the preserved $prec_k^\pi$ formulas.

It is clear from the above definition that logical reasoning is necessary for the detection of subgoal ordering constraints. Such a requirement may make computation infeasible in a complicated problem domain. Therefore, to make this approach feasible, it is necessary to impose the following assumption: *In problem operator specifications, everything implied by $post_k$ ($prec_k$) is also explicitly represented in $post_k$ ($prec_k$).*

Theorem 4

$$\forall k \forall \pi \left[(con(post_k^\pi, \wedge_i g_{k_i} \wedge_j g_{h_j}) \longrightarrow (Prec_k^\pi \longrightarrow \wedge_i g_{k_i})\right] \wedge$$
$$\forall g \in \{g_{h_j}\}\{\exists k \exists \pi \left[(con(post_k^\pi, \wedge_i g_{k_i} \wedge g) \wedge \neg(Prec_k^\pi \longrightarrow g)\right]\}$$

if and only if

$$\forall k \forall s (T_k(s) \in S_{\wedge_{i,j} g_{k_i}, g_{h_j}} \longrightarrow s \in S_{\wedge_i g_{k_i}}) \wedge$$
$$\forall g \in \{g_{h_j}\} \exists k \exists s (T_k(s) \in S_{\wedge_i g_{k_i}, g} \wedge s \notin S_g)$$

where $Prec_k^\pi$ is the conjunction of $prec_k^\pi$ and those subgoals of G which do not match anything in $post_k$.

This theorem provides an equivalent membership criterion for the relation $\prec^+$. The new criterion leads directly to the procedure, $ONECON$, for detecting subgoal ordering constraints.

```
Procedure:     ONECON{GS,C)
Input: GS———a set of subgoals;
Output: C———a member of $\prec^+$ over $2^{GS}$.
fail— $\phi$;
For each problem operator schema $r_k$ do:
  Begin {For $r_k$}
    For each binding $\pi$ of $post_k$ with GS do
      If $con(post_k^\pi, GS)$ then
        Begin {If}
          $GS' = GS - fail$
          For each $g \in GS'$ do
            if $Prec_k^\pi \longrightarrow g$
              then $pass_{k,\pi} = pass_{k,\pi} \cup \{g\}$;
              else $fail = fail \cup \{g\}$;
        End{If};
    $pass_k = \cap_\pi pass_{k,\pi}$;
  End{For $r_k$};
$pass = \cap_k pass_k$;
if $pass = \phi$
  then Return $\phi$
  else Return($< pass, GS - pass >$).
```

Based on the assumption made on problem operator representation, the complexity of ONEGON is linear in n. The upper bound for the worst case complexity of GEN CON is then $0(n^3)$.

[3] An example of a problem constraint is: *no problem object can ever reside in two rooms simultaneously* .

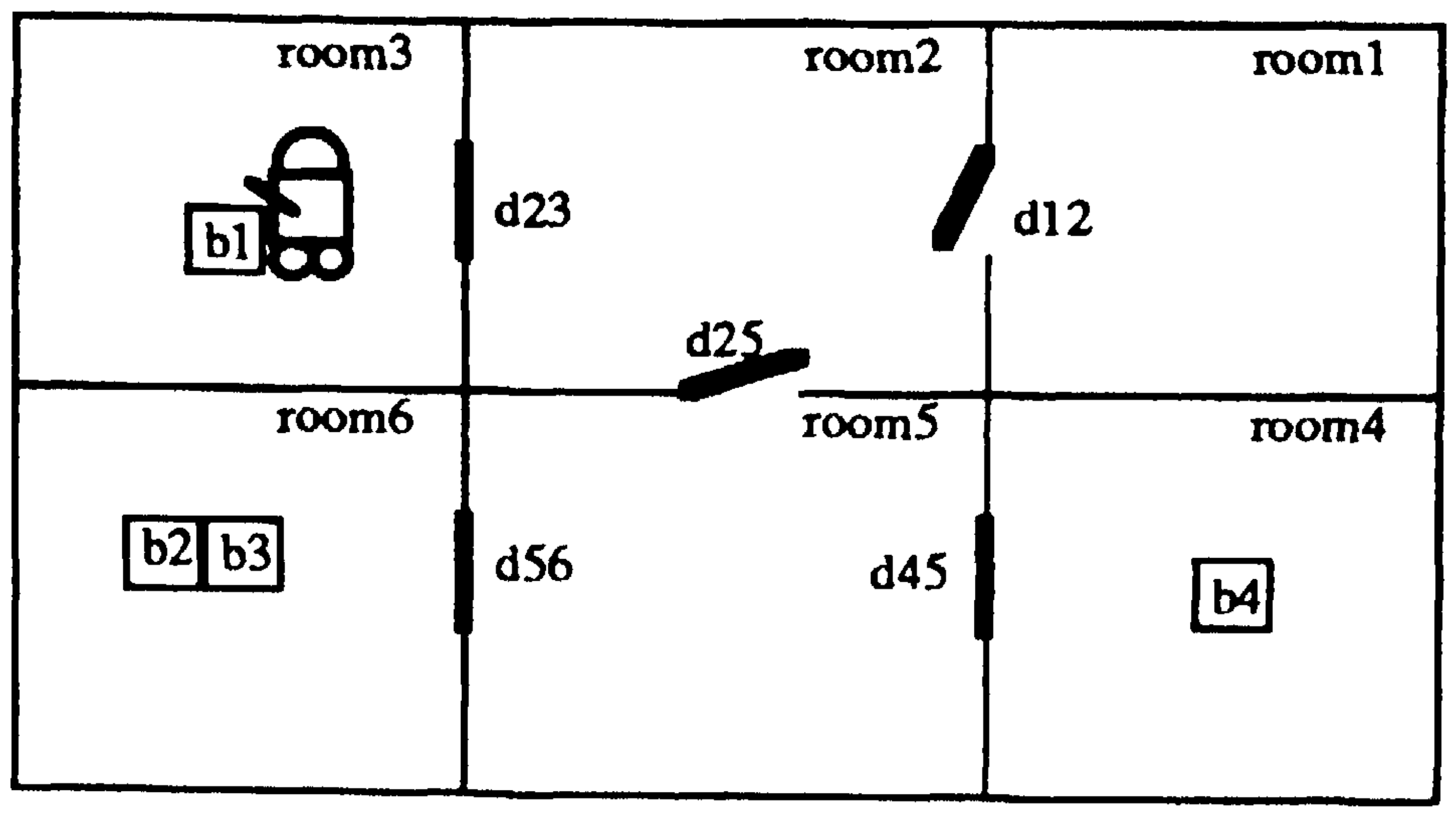

Figure 1: A Problem Goal State

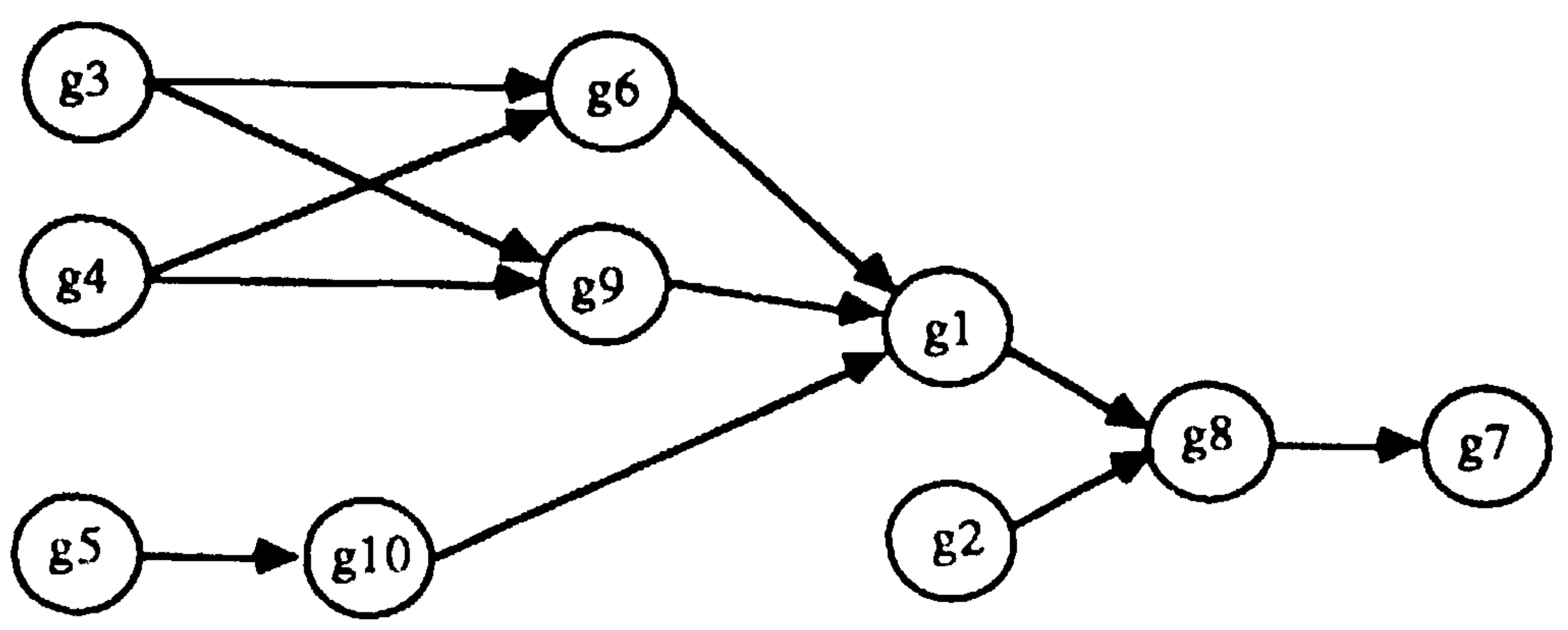

Figure 2: Subgoal Ordering Constraints

4 An Example

In this section, we use a simple robot planning example to illustrate our approach. The problem involves one robot, four blocks, six rooms and five doors. The problem operators are chosen from those used by Sacerdoti [1974]. The names of these operators are. $GOTOB(bx)$, $GOTO(dx)$, PUSHB (bx, by), PUSHD(bx, dx) $OPEN(dx)$, CLOSE(dx), GOTHRUDR(dx, rx) and PUSHTHRUDR (bx, dx, rx). The specifications for these operators are not listed here because of space restriction. A possible goal state is shown in Figure 1 and the problem subgoals comprising the problem goal are given as follows:

(g_1) : $Inroom(robot, r3)$;	(g_2) : $Inroom(b1, r3)$
(g_3) : $Inroom(b2, r6)$;	(g_4) : $Inroom(b3, r6)$;
(g_5) : $Inroom(b4, r4)$;	(g_6) : $Nextto(b2, b3)$;
(g_7) : $Nextto(robot, b1)$;	(g_8) : $closed(d23)$;
(g_9) : $closed(d56)$;	(g_{10}) : $closed(d45)$;

After applying the procedure $GENCON$, we get

$$R_{min} =$$
$$\{< g_1 \wedge g_2 \wedge g_3 \wedge g_4 \wedge g_5 \wedge g_6 \wedge g_8 \wedge g_9 \wedge g_{10}, g_7 >,$$
$$< g_1 \wedge g_2 \wedge g_3 \wedge g_4 \wedge g_5 \wedge g_6 \wedge g_9 \wedge g_{10}, g_8 >,$$
$$< g_3 \wedge g_4 \wedge g_5 \wedge g_6 \wedge g_9 \wedge g_{10}, g_1 >,$$
$$< g_3 \wedge g_4 \wedge g_6 >, < g_3 \wedge g_4, g_9 >, < g_5, g_{10} >\}$$

From R_{min}, the constraint graph can be generated as shown in figure 2. In the figure, a directed arc represents the constraint that the subgoal in the source node must precede the subgoal in the destination node for this problem.

5 Conclusion

The main issue brought up by this research is that an explicit representation of subgoal ordering constraints can greatly facilitate the development of subgoal ordering strategies. The representation makes it clear what type of constraints one is dealing with and how the constraints can be detected by analyzing problem specification. Based on that representation, many generic properties of the constraints can be easily inferred. Furthermore, the representation helps one to understand the capabilities of the subgoal ordering approaches.

References

[Chapman, 1987] Chapman, David, "Planning for Conjunctive Goals", *Artificial Intelligence 32,* 1987. pp. 333-377.

[Cheng, 1989] Cheng, Jie, "A Systematic Approach to Problem Subgoal Ordering", Ph.D. thesis in preparation, EECS Department, University of Michigan, Ann Arbor, Michigan. 1989.

[Dawson and Siklossy, 1977] Dawson, C, and Siklossy, L., "The Role of Preprocessing in Problem Solving Systems", *Proc.IJCAI 5,* Cambridge, Mass., August 1977.

[Ernst and Goldstein, 1982] Ernst, G. W., and Goldstein, M. M., "Mechanical Discovery of Classes of Problem Solving Strategies", *JACM,* Vol. 29, No. 1, January 1982. pp. 1-23.

[Irani and Cheng, 1987] Irani, Keki B. and Cheng, Jie, "Subgoal Ordering and Goal Augmentation for Heuristic Problem Solving", *The Proceeding of the 10th IJCAI,* 1987, pp. 1018-1024.

[Sacerdoti, 1974] S acerdoti, E. D., "Planning in a Hierarchy of Abstraction Spaces", *Artificial Intelligence 5,* 1974. pp. 115-135.

[Sacerdoti, 1975] Sacerdoti, E. D., The Nonlinear nature of plans, *Advance Papers IJCAI-75,* Tbilisi, USSR. 1975. pp. 206-214.

[Sacerdoti, 1977] Sacerdoti, E. D., *A Structure for Plans and Behavior,* American Elseview, New York, 1977.

[Sussman, 1975] Sussman, G. J., *A Computational Model of Skill Acquisition,* American Elseview, New York, 1975.

[Tate, 1975] Tate, A., "Interacting Goals and Their Use", *The Proceeding of the 4th IJCAI,* 1975, pp. 215-218.

[Waldinger, 1981] Waldinger, R., "Achieving Several Goals Simultaneously". *Readings in Al,* 1981. pp. 250-271.

[Warren, 1974] Warren, David H. D., "WARPLAN: A System for Generating Plans", Department of Computational Logic Memo 76, U. of Edinburgh, July, 1974.