A Symbolic Approach to Qualitative Kinematics

Boi Faltings
Laboratoire d'Intelligence Artificielle
Departement d'Informatique
Swiss Federal Institute of Technology (EPFL)
MA-Ecublens
1015 Lausanne
Switzerland

Tel. +41-21-693-2738
E-Mail: faltings@elma.epfl.ch

Draft of February 21, 1992

Abstract

An important problem for mechanism design and analysis is reasoning about the relationship between object shapes and their kinematic function. Such reasoning is difficult because of the unstructured influence of the shapes' metric dimensions. In this paper, we show how a qualitative kinematic analysis can be based solely on symbolic reasoning and evaluation of predicates on metric dimensions. This allows symbolic reasoning about kinematics without explicit numerical representations of object dimensions, and automatic generation of operators relating kinematic goals to shape modifications which may achieve them.

1 Introduction

Reasoning about object shapes and their interaction is an important unsolved problem in Artificial Intelligence. Mechanisms such as the mechanical ratchet shown in Figure 1 are good examples of complex interactions of shapes, and can not be reasoned about using traditional theories of qualitative physics ([DKL75, HAY79, FOR84]).

INSERT FIGURE 1 ABOUT HERE

The ratchet consists of two parts, a wheel and a pawl, both of which are hinged at fixed axes. Its function, to restrict motion of the wheel in the clockwise direction while freely allowing it in the counterclockwise sense, is achieved by the interaction of the shapes of its parts, its kinematics. In earlier work on *qualitative kinematics* ([FNF91, FALT87a, FALT90]) we have developed methods for the qualitative *representation* of kinematic function. In this paper, we investigate the task of first-principles reasoning about how changes in the objects' shapes affect this kinematic function.

As an example of the type of reasoning we are addressing in this paper, consider the pair of gearwheels shown in Figure 2. People can readily state some conditions that the distance between their centers must satisfy in order for the gears to mesh.

INSERT FIGURE 2 ABOUT HERE

This type of reasoning is important in many spatial reasoning problems including mechanical design, troubleshooting, learning new physical phenomena, and reasoning under uncertainty. The

Figure 1: A ratchet.

Figure 2: A pair of gears meshes only when the distance d between their centers is a) small enough for the gears to touch, and b) large enough so that the teeth fit into each other. Note that these conditions are necessary but not sufficient - the shapes of the teeth must also be compatible, for example.

fact that people can state conditions for behavior with such apparent ease is surprising and not explained by any existing theory of spatial reasoning.

In this paper, we propose a computational theory of such reasoning based on the idea of *inverting* dependencies between shape features and states of the kinematic behavior. Object shapes are represented by a *metric diagram*, in which a metric for the shapes' dimensions is defined as a set of lumped parameters. The dependency of kinematic function on the shapes is given by metric *predicates* defined in terms of these parameters, similar to the conditions shown in Figure 2. In the kinematic analysis, all access to object dimensions is formulated in terms of metric predicates. Every kinematic property is marked with the metric predicates and thus the metric parameters it depends on. This provides a basis for reasoning about the relation between shape and kinematic function.

The predicates could be constructed and evaluated by independent parallel processes, and on a massively parallel computer the computation could actually be faster than a sequential numerical simulation. Besides allowing explicit reasoning about the relation of shape and kinematic function, the approach presented in this paper could therefore provide powerful methods for qualitative kinematic analysis.

1.1 Reasoning about shape and function

The most difficult problem for reasoning about the relationship of shape and function is the influence of the metric object dimensions on kinematic function. Because the dimensions of physical objects are not naturally defined as scalar quantities, we represent all information about the objects and their shapes in a *metric diagram*, an example of which is shown in Figure 3.

INSERT FIGURE 3 ABOUT HERE

The metric diagram consists of three parts:

- a symbolic representation of the boundaries of each of the objects, defining a symbol for each metric dimension. This is similar in sprit to a primal sketch in computer vision ([MARR81]).
- a representation of knowledge about the metric dimensions. In Figure 3, this is simply a list of approximate numerical values. However, it could also be a set of intervals, or even a measurement procedure which only determines a value when needed, similar to the idea of visual routines ([ULL84]).
- additional annotations, such as the freedom of motion of an object or the fact that a shape is periodic.

The metric diagram separates the symbolic representation of the shapes and their dimensions from that of the actual numerical values of the metric parameters. However, this is not yet sufficient for applying the traditional formalisms of qualitative physics to kinematic analysis, because sign or interval representations of metric dimensions do not express useful kinematic distinctions. Useful qualitative categories are established by metric *predicates*, where the qualitative distinction is made according to the sign of a nonlinear expression involving the individual dimensions. This

Figure 3: The metric diagram for the ratchet example.

is the key idea which makes general symbolic reasoning about qualitative kinematics possible. Based on the values of these predicates and the symbolic description, a qualitative kinematic analysis can be computed by purely symbolic reasoning.

Metric predicates are evaluated by an abstract access procedure, which could be implemented in a variety of ways: numerical calculation, measurements in an image, or even experiments on the objects. When the information is not sufficiently precise to determine the predicate value, kinematic function can be predicted based on assumptions about the predicate value.

The most important aspect of structuring the computation in this way is the fact that the metric predicates constitute an explicit representation of the influence of metric parameters on kinematic function. A predicate which is used to predict a particular aspect of kinematic function can be interpreted as a *condition* for this function to exist. Just as for people, it is now easy to derive conditions such as those in Figure 2.

The conditions can be used for reasoning in two ways. First, they indicate which of the metric dimensions influence a particular kinematic function. Second, they provide a problem-dependent qualitative representation of the continuous space of metric dimensions. Variation of a metric dimension has an influence on kinematic behavior only through the predicates that it occurs in. We use the term *landmark values* for the values of a metric dimension where a predicate changes its value. The complete set of landmark values for a metric dimension defines a *quantity space* ([FOR84]) within which it can be qualitatively represented. However, in contrast to quantity spaces as defined in qualitative process theory ([FOR84]), these landmark values depend on the values of the other system parameters. An important condition for the validity of such a quantity space is that the predicate formulation itself is independent of the actual value of the dimension which is being represented. We show in this paper that such independent predicates[1] can be found for the domain of elementary kinematic interactions.

The important aspect of the metric diagram model is that it allows a clean symbolic treatment of metric information. This provides an alternative (and, on parallel computers, more efficient) way of analyzing kinematics, but more importantly allows reasoning about the dependencies between shape and kinematic function. Such reasoning is important for problems such as intelligent computer-aided design, mechanism verification and diagnosis, and machine learning about kinematics.

1.2 A qualitative functional analysis of the ratchet

There are two main qualitative characteristics that can be ascribed to a particular configuration of a mechanism:

- the points of contact between its parts
- the way that forces and motions are transmitted by the contact points

We define a *place vocabulary* of a mechanism to be the graph of states which describe the different possible values that these characteristics can take on, and the possible transitions between them.

[1] With some minor exceptions, for which there is nevertheless a solution using algebraic decision methods.

Sample configurations of some of the states in the place vocabulary for the ratchet in Figure 1 are shown in Figure 4.

INSERT FIGURE 4 ABOUT HERE

The place vocabulary forms the basis for an *envisionment* of the qualitative *behavior* of the device under external influences. When particular qualitative motions and forces are given, the complete behavior of the device can be inferred using the information encoded in the place vocabulary. As an example, the function of the ratchet can be explained using the states in Figure 4 as follows.

Under the force of gravity, the pawl is forced onto the wheel. Among the configurations where it rests on the wheel, we can distinguish those where it points to the right from those where it points to the left. When the lever points to the right, and rotation is counterclockwise, we find the sequence of states e) $\rightarrow$ f) $\rightarrow$ e) $\rightarrow \ldots$, repeated for each tooth. When the lever points to the left, and the wheel again turns counterclockwise, the behavior of the ratchet is characterized by the sequence of states a) $\rightarrow$ b) $\rightarrow$ c) $\rightarrow$ a) $\rightarrow \ldots$, also repeated for each tooth. Now, consider the case where the wheel turns clockwise. With the lever pointing to the right, the sequence of states is f) $\rightarrow$ e) $\rightarrow$ f) $\rightarrow \ldots$, the reverse of the counterclockwise sequence. However, when the lever points to the left, the sequence of motions is b) $\rightarrow$ a) $\rightarrow$ d). From state d), there is no further transition in which the wheel could turn clockwise; the ratchet is blocked. When the lever is in a configuration where it points to the left, the ratchet thus blocks turning the wheel in a clockwise direction, but does not affect its motion in the counterclockwise direction. This is the

Figure 4: Examples of possible states of the ratchet.

kinematic function of the ratchet.

The place vocabulary which allows such reasoning is a graph representing a region decomposition of the device's *configuration space*, the space spanned by the position parameters of the parts. Each point in the configuration space represents a particular configuration of the parts, and the complicated geometric constraints on their motion are reduced to constraints on the motion of a point. Configuration space was invented by Heinrich Hertz to formalize the laws of mechanical motion, and has since been widely used as a theoretical tool by physicists and mechanical engineers. A similar formalization used in theoretical kinematics is the *soma* of Study, which differs from configuration space in the coordinates used to describe the configuration.

As an explicit computational device, configuration space has been used as a tool for robot motion planning ([LPW79, LMT83, SHS83]). The computational techniques developed by these researchers have been combined with the place vocabulary idea of Forbus ([FOR80]), resulting in the place vocabulary theory of qualitative kinematics ([FALT90]) which has been used for qualitative simulation of mechanisms such as a clock ([FNF91]). In contrast to the theories developed in theoretical kinematics, the theory of qualitative kinematics is a computational theory whose main focus is to provide a functional representation which is efficiently computable. In this paper, we show how this computation itself can be exploited for reasoning explicitly about the relation between object shapes and kinematic function. Such reasoning about how changes in object geometries affect the configuration space have not been investigated so far by other researchers using configuration space. For example, in robot motion planning this would correspond to reasoning about changing the object or robot geometry to make certain motions possible or impossible. It is likely that the methods developed in this paper could be generalized to such applications, but we have not investigated this possibility yet.

1.3 Limitations

The methods presented in this paper are subject to the same limitations as the place vocabulary theory of qualitative kinematics itself ([FALT90]). The first restriction, which is essential to the algorithms underlying the theory, is that in handles interactions of two objects with one degree of freedom each. However, we are currently working on techniques which will allow generalization to interactions involving more degrees of freedom.

The second restriction, which can be relaxed without fundamental changes to the methods presented in this paper, is that the object shapes have to be two-dimensional, with boundaries consisting of straight lines and circular arcs.

While the restrictions appear severe, they still allow the theory to handle a large set of interesting mechanisms, such as certain mechanical clocks ([FALT90]). Furthermore, the methods proposed in this paper may be applicable to many other domains, and mechanism kinematics is just an example of their usefulness.

2 Qualitative Representation of Metric Dimensions

Restricting the use of the metric object dimensions to the evaluation of predicates makes it possible to reason about their influence on kinematic behavior by analyzing the predicate expressions. The metric dimensions which determine a particular kinematic function are identified by the set of predicate expressions that are used in the computation of the places corresponding to the function.

On the other hand, variation of metric dimensions can influence kinematic function only if it changes the value of one of the predicates involved. Let the *metric space* be the (many-dimensional) space spanned by all the metric dimensions defined in the metric diagram, and let $\mathbf{x}$ be a point in metric space, i.e. a vector of all the metric parameters. Each predicate can then be characterized as the test of the sign of an expression $\mathcal{C}(\mathbf{x})$. The curve where $\mathcal{C}(\mathbf{x}) = 0$ divides metric space into two or more regions distinguished by different values of the predicate. Now consider the complete set of predicates used in the computation of the place vocabulary for a particular device. It divides metric space into a set of regions such that within each region, the values of all the predicates - given as the signs of the algebraic expressions - are constant. An example of such a division for a (hypothetically simple) two-dimensional metric space is illustrated in Figure 5. Because kinematic function is completely determined by the set of all predicate values, this region structure defines *all* meaningful qualitative distinctions of metric object dimensions which can be made for the shapes. In principle, it would be possible to represent the metric dimensions of objects qualitatively in this region structure.

In a practical case, the metric space has a very large number of dimensions. The shape of

Figure 5: Metric space is subdivided into qualitatively equivalent cells by curves corresponding to the predicate expressions. When only a single parameter is variable, the cells are projected onto *intervals* separated by *landmark values*, as shown here for x_1.

a simple ratchet wheel has at least three independent parameters (radius and the lengths of the two sides of the teeth), and at least three are required to specify the dimensions of the pawl. More realistically, metric parameters are defined as unknown coordinates in the metric diagram, resulting in 18 parameters for the ratchet. Moreover, in general it is not possible to define a complete set of truly independent parameters, so that metric space is non-Euclidian. All these features make the explicit computation of the region structure of metric space appear to be a very hard problem. However, existing algebraic cell decomposition techniques ([BKR85, KY85]) can compute this region structure, but they are extremely inefficient.

The problem is simplified if we consider only subsets of the metric dimensions, in particular single parameters. The projection of the complete region structure onto the axis of a single dimension is given as a set of *landmark values* of this parameter. Rather than by explicit projection of the regions in metric space, the landmark values can be computed directly as roots of the predicate expressions using well-developed and efficient techniques of numerical analysis, and often closed-form solutions are possible.

2.1 Enumeration Analysis

A sampling of the metric space along a single parameter axis defines the set of intervals for this parameter within which the place vocabulary is constant. A set of sample place vocabularies for each interval thus defines a complete *enumeration* of all possiblities that can be achieved by varying the parameter. An example of such an *enumeration analysis* for the ratchet shown earlier in Figure 1 is illustrated in Figure 6. The parameter which is varied in Figure 6 is the distance between the centers of rotation of wheel and pawl, represented by the parameter Y0-LEVER in Figure 3. Restricted to the interval between 30 and 80[2], the system finds a total of 14 landmark values, at 51.0045, 52.479, 53.1375, 58.30565, 59.0998, 59.2725, 59.28925, 59.38427, 60.72458, 61.62136, 62.0, 67.7359 and 70.28427. Below the landmark value at 51.0045, there exist no legal configurations of the objects, and beyond the landmark value at 70.28427, no contact between the objects is possible. The changes that become apparent as Y0-lever varies between these bounds are illustrated in Figure 6.

INSERT FIGURE 6 ABOUT HERE

A symmetric set of landmark values exists for placements of the lever below the wheel, in the interval between 8.9955 and -10.28427, but are not considered because of the restriction to the interval between 30 and 80. The most significant changes in behavior occur at 59.38427, where it becomes possible for the wheel to turn, and at 62.0, where the wheel can turn freely in both directions. Choices of Y0-LEVER in the interval between 59.38427 and 62.0 result in a functioning ratchet. The other landmark values reflect changes in possible contact relationships and qualitative inference rules associated with places.

The fact that there are only 14 different landmark values for the variation of a parameter as central as the distance between the centers of rotation is a surprising fact. The total number of predicates considered in this example is far higher, namely 10278. However, only very few of

[2] A bounding interval is imposed so that roots can be found by binary search.

Figure 6: Enumeration analysis lists all the changes in the ratchet place vocabulary as Y0-LEVER increases from 30 to 80.

these predicate expressions actually have roots in the given interval. Because of the periodicity of the device, many of the remaining roots actually fall on the same value. Finally, several landmark values reflect changes in unreachable configurations, resulting in the same place vocabulary, and can also be eliminated. In the current program, this is detected by explicit computation of the place vocabulary. However, it would also be possible to explicitly maintain the dependencies between landmark values and the parameter intervals where they are applicable.

Enumeration analysis has numerous applications in mechanism design and analysis. It can also be useful for modeling devices with variable geometries, such as that shown in Figure 7.

INSERT FIGURE 7 ABOUT HERE

The large lever in Figure 7 can either leave the wheel completely free to rotate, enable rotation in one direction only, or block rotation completely. We assume that the position of the large lever is fixed by some device and only changed by the operator of the device. It functions as a switch for selecting different dynamic behaviors of the mechanism, and changing its setting is not part of the normal duty cycle. It should therefore not be modelled as part of the dynamic behavior, but as a condition for selecting the appropriate device model. This is the idea of a graph of models proposed by Penburthy and Addanki ([PEN87, ADA87]). Enumeration analysis can find or verify this graph of models.

Figure 7: The ratchet is either disengaged, active, or blocks the wheel completely, depending on the position of the lever.

3 Maintaining Dependencies

Enumeration analysis provides the set of landmark values required for reasoning about the influence of a particular parameter on kinematic function, but does not help with identifying which of the many object dimensions are responsible for a particular aspect of this function. This identification is important for kinematic problem solving, where the goal is to achieve or modify a particular aspect of the kinematic function. It can be achieved by maintaining an explicit dependency structure relating object dimensions, evaluated predicates and elements of the place vocabulary, similar to a reason maintenance system ([DOY79]).

We assume that desired modifications or explanations of kinematic behavior are stated in terms of the place vocabulary as either places to be added or eliminated, or as changes in the characteristics of places or transitions. In the computation of the place vocabulary, each such element is marked with the predicate expressions and values it depends on. Each dependency is marked with its particular *role*, distinguishing predicates that determine the existance of a place from those that determine different aspects of its form.

In contrast to the dependencies used in reason maintenance systems ([DOY79]), which give *sufficient* conditions for a conclusion, the metric predicates attached to a particular place are necessary, but not sufficient conditions[3] for achieving the particular function. The reason for this is that it is usually possible to add or change completely unrelated shape features and thereby

[3]With the possible exception of some predicates which determine local characteristics, which could be considered both necessary and sufficient.

create new interactions which *subsume* the place in question. This allows eliminating the place while still satisfying all its conditions, which therefore cannot be sufficient. As this argument applies independently of the particular way the conditions are defined, it means more generally that it is impossible to formulate sufficient conditions for the presence of a kinematic function.

Given a particular place which currently exists in the place vocabulary, the necessary conditions for its existance in this form define the possible local shape modifications which would eliminate or modify it. For each predicate, the set of parameters required for its evaluation in turn defines the object dimensions whose variation could effect the modification.

3.1 Perturbation Analysis

Besides determining *which* parameters to change, it is also possible to calculate *how much* a parameter has to change to affect kinematic behavior. Given that all other parameters remain constant, the variation of the chosen parameter has to be sufficient to change the value of the predicate. This is the case only if it is perturbed at least beyond one of the closest roots of the predicate expression. We call such an analysis of possible local changes of parameters a *perturbation analysis* of the device. Note that this perturbation analysis is different from that of Daniel Weld ([WEL88]), who uses the term for the analysis of the differential relation between system parameters. His analysis is complementary because it is based on continuous variations, whereas our methods address discontinous, qualitative changes of behavior.

By such inversion of dependencies, it is possible to construct a set of specialized operators for kinematic problem solving. As an example, consider the ratchet shown in Figures 1 and 3 in the introduction, and the task of changing the device to suspend the ratchet function. This requires eliminating the state where the pawl blocks rotation of the wheel, represented in the place vocabulary as the place named IP-2 and shown in Figure 8.

INSERT FIGURE 8 ABOUT HERE

The perturbation analysis for IP-2, shown in Figure 9, shows that two predicates were used in predicting the existance of IP-2.

INSERT FIGURE 9 ABOUT HERE

Picking the first of the two predicates for modification[4], we find that it involves the following parameters:

- The placement of wheel and lever in the workspace, described by parameters Y0-WHEEL, Y0-LEVER, X0-WHEEL and X0-LEVER.
- The coordinates XL2 and YL2 of the tip of the lever
- The coordinates XW1 and YW1 of the bottom of the teeth of the wheel

By perturbation analysis, it is possible to compute specialized operators for modifying each of these parameters. An algebraic simplifier substitutes the actual values for all except the chosen

[4] It turns out that the value of the second does not change within the range of feasible modifications of object dimensions.

Figure 8: The configuration where the pawl blocks the wheel, described by place IP-2.

Figure 9: Ouput of the program for perturbation analysis of IP-2.

parameter and simplifies the expression. An approximation of the closest landmark value is then found by binary search within a user-specified interval[5].

Three examples of this process are shown in Figure 9. One possible way to eliminate the ratchet function is to place wheel and pawl further apart, for example by variation of Y0-LEVER (see Figure 3 for the precise parameter definitions). Perturbation analysis finds a landmark value at Y0-LEVER= 62 (slightly inaccurate because of the numerical approximation), which is the value where the tip of the lever can just touch the bottom of the wheel. Choosing a greater value will eliminate the place IP-2. Other possible variations are the length of the lever, by changing the value of XL2 beyond the landmark value at XL2=10, and the coordinates of the bottom of the wheel's teeth, by changing YW1 beyond the landmark value of 18. Both landmark values again correspond to a design where the tip of the lever can just touch the bottom of the wheel's teeth.

Many design modifications require changing only local details of the place vocabulary. In the ratchet example, the state where the side of the pawl rests on the wheel exists in two different versions, one in which the counterclockwise motion of the wheel pushes the pawl up (turning it clockwise) and one where it allows it to descend (turning it counterclockwise). A configuration corresponding to this latter state is shown in Figure 10.

INSERT FIGURE 10 ABOUT HERE

In this state, represented in the place vocabulary by a place called CSEG-17, the wheel can be

[5] This process could be improved by using more advanced numerical analysis techniques.

Figure 10: The lever, pulled down by gravity, pushes the wheel in the counterclockwise direction.

driven forward (counterclockwise) by a positive moment on the pawl. If it is linked to a gear transmission, the play between the gears leads to unncecessary rattle and wear. Thus, we might want to eliminate this state from the place vocabulary.

There are two predicates associated with the existance of CSEG-17. It turns out that the landmark values for the first of them correspond to radical changes in the place vocabulary and are not useful for local modifications. Perturbation analysis of the parameters in the second predicate results, for example, in the following modifications:

- Shorten the pawl, by varying XL2 beyond the landmark value of 6.975.
- Decrease the height of the teeth, by lowering YW2 below 26.5248.
- Increase the distance between the objects, by increasing Y0-LEVER beyond 61.62135.

Of these suggested operators, shortening the pawl also eliminates the blocking state crucial to ratchet function. The other two, however, have exactly the desired effect, without any other modifications to the place vocabulary.

As a final example, we might want to change the way that the objects touch each other in a particular configuration. In the ratchet example, it is possible for the tips of the wheel's teeth to touch the sides of the pawl, as shown on the left in Figure 11. Suppose we wanted to avoid this type of touch, instead letting the tip of the pawl touch the wheel, as shown in the center of Figure 11.

INSERT FIGURE 11 ABOUT HERE

Figure 11: In the state on the left, the tips of the wheel's teeth touch the side of the pawl. Another possible touch in a similar configuration is the one shown in the middle. Which situation is present depends on the situation at IP-12, shown on the right.

Depending on the situation at the place IP-12, shown on the right in Figure 11, one of the two types of touch is more constraining and subsumes the other. The selection between the two is tested by one of the predicates which determine the adjacencies between IP-12 and neighbouring places. One of the metric parameters involved in this predicate is YW2, expressing the height of the wheel's teeth. Perturbation analysis finds a landmark value of 21.7328, and indeed it turns out that changing YW2 accordingly results in a functional ratchet in which only this particular type of touch has been replaced.

In each of the examples shown, perturbation analysis has found at least one operator which actually achieves the goal. However, success can not be guaranteed: a mechanism with the desired characteristics may simply not exist, or may require simultaneously changing several parameters. With some exceptions, such as that of the last example, it is also impossible to *add* functions to the place vocabulary, as dependencies are only constructed for already existing functions. Using perturbation analysis to add functions would require local conditions for their *absence*. It would be very surprising if such conditions existed: functions can be added by shape features elsewhere on the mechanism, independent of any local conditions for their absence.

4 Kinematic Analysis using the Metric Diagram

The methods for symbolic reasoning about kinematics presented in the previous sections depend crucially on the fact that the qualitative kinematic analysis can be based on metric predicates. In this section, we discuss the metric predicates required for computing place vocabularies in the metric diagram framework. For lack of space, the discussion can not be exhaustive. A full and detailed description of the algorithms is found in [FALT87b]. We also limit the discussion to the kinematic interaction of a pair of two-dimensional objects, both of which have *rotational* freedom. The same methods and considerations apply, with minor changes, to interactions of objects with translational freedom ([FALT87b]).

4.1 Configuration space representation

The *configuration space* (c-space) of a mechanism is the space spanned by the degrees of freedom of its parts. Each configuration of the mechanism corresponds to a point in c-space; if the configuration is *legal*, i.e. free of overlaps, we say that the corresponding c-space point falls within *free space*, otherwise it lies in *blocked space*. In our case of mechanisms consisting of two parts with one rotational degree of freedom each, the configuration space is a torus surface, spanned by the orientations of the two parts.

Kinematic function is achieved by the fact that configurations of the device are restricted to free space, i.e. constrained by the boundary between free and blocked space. Configurations on this boundary are characterized by the condition that the objects are in contact. With the object boundaries restricted to arcs and straight lines, the only possible contacts are (i) between a vertex on one object and a boundary on the other, and (ii) between a circular arc boundary and another boundary segment. The two types of touch are illustrated in Figure 12.

INSERT FIGURE 12 ABOUT HERE

Figure 12: Types of touch defining vertex constraints (left) and boundary constraints (right). Note that the condition defining a boundary constraint amounts to that of a vertex constraint in which the center of the arc touches an imaginary offset boundary.

Each of the possible contacts defines a *constraint* in configuration space as the set of configurations where the objects touch at the point in question. We call the constraints generated by a vertex contact *vertex constraints* and those generated by a boundary contact *boundary constraints*. The boundary between free and blocked space, which is responsible for kinematic function and represented by the place vocabulary, is given as the *envelope* of the set of constraints.

The algebraic equation of the curve of a vertex constraint is obtained by substituting the coordinates of the vertex in the equation of the boundary segment, where both coordinates have been transformed into a common global coordinate system. As this transformation involves the configuration space parameters, the result is a curve equation in two parameters: $\mathcal{C}(\phi, \psi) = 0$. Boundary constraints for circular arcs are obtained using the analogy to a vertex constraint with an imaginary boundary shown in Figure 12.

For a particular pair of objects, the symbolic part of the object descriptions is sufficient to define the finite set of possible constraints: each pair of a boundary segment of one object and either a vertex or a curved boundary on the other defines a constraint. Because constraints are algebraic curves, there are only a finite number of ways they can intersect each other, and thus a finite number of envelopes and place vocabularies they can form. The role of metric information in the computation of place vocabularies can be understood to be the *selection* of the correct place vocabulary using a set of metric *predicates*. The fact that the metric predicates are responsible for selecting the correct solution is fundamental to the methods presented in this paper.

However, explicit selection among the possibilities is not possible because their number is far too large: Even for a pair of 2 rotating triangles, there are 18 possible constraints, up to $\frac{18(18-1)}{2} = 153$ possible intersections between constraints, and therefore up to 2^{153} possible intersection patterns! Metric information must be used incrementally during the computation to avoid generating such a huge number of possibilities. However, the predicates this computation uses are exactly those that would be required for a selection among explicitly generated possibilities[6], and therefore satisfy the assumptions underlying the reasoning methods.

Metric predicates can be grouped into three categories, for determining the set of active c-space constraints, the local properties of constraints, and free subsumptions.

4.2 Indexing points of touch

Many of the metric predicates refer to particular touch configurations, best described by a parameterization of constraints. Consider thus the touch configuration of a pair of points on two plane objects which are free to rotate around fixed centers, as shown in Figure 13.

INSERT FIGURE 13 ABOUT HERE

The dashed lines in Figure 13 indicate the location of fixed "zero-directions" on the objects, and the angles ψ_1 and ψ_2 between them and the (fixed) connection between the centers of rotation are the two configuration space parameters. ψ_1 and ψ_2 can be calculated using the five geometric parameters:

[6]However, in the analysis of free subsumptions, more predicates than really necessary are generated.

Figure 13: Configuration of touch of points P_1 and P_2 on two rotating objects.

- the distance d between the centers of rotation, a constant.
- the distances r_1 and r_2 of each of the points from the center of rotation of its object, called the *radius*.
- the offset angles ϕ_1 and ϕ_2 between the radius vector to each point and the fixed "zero-direction" on its object.

Both the radius and the angle ϕ can be used for designating the point on an edge (and thus on the constraint) where the touch occurs. Most metric predicates are formulated in terms of radius values, and we therefore parameterize constraints by radius. The offset angle is given by a function depending on the form of the edge. Because an edge may contain several points with the same radius, it is often necessary to break edges in half[7] to make the identification unambiguous[8]. We assume that edges are broken up at radius extrema at the level of the symbolic description, so that the relation between offset angle and radius is always a function. Using this parameterization, for any constraint and radius value, we can compute the configuration of touch in closed form, provided it exists (as shown in [FALT87b]).

For any touch between a pair of points on rotating objects there exists a *mirror image* symmetrical to the axis between the two centers of rotation, as indicated by the dotted lines in Figure 13. For this reason, any configuration (and any constraint) exists both as an *original*,

[7]Up to three parts for arcs.

[8]This might seem a disadvantage of this parameterization. However, we shall see that there are additional, independent reasons why this breakup is required.

for which the touch is to the right of the vector between object 1 and object 2, and a *mirror image*, for which it falls on the opposite side.

4.3 Selecting active constraints

A vertex constraint generated by a vertex V and and edge E can have an influence on the kinematics only if there actually exist configurations where V touches E, in which case we call the constraint *active*. Letting r_V stand for the radius of V and d for the distance between the centers of rotation, V can touch a point on E with radius r if the two following conditions are satisfied:

$$d < r_V + r \text{ and } d > |r_V - r|$$

If the edge E contains points whose radii lie in the interval $[r_{min}..r_{max}]$, there is at least one such configuration whenever the two conditions are satisfied by one of r_{min} or r_{max}. This defines the condition for the constraint to be active:

$$(\mathcal{P}_1 \vee \mathcal{P}_2) \wedge (\mathcal{P}_3 \vee \mathcal{P}_4) \quad \text{, where}$$

$$\mathcal{P}_1 = (d < r_V + r_{max})$$

$$\mathcal{P}_2 = (d < r_V + r_{min})$$

$$\mathcal{P}_3 = (d > |r_V - r_{max}|)$$

$$\mathcal{P}_4 = (d > |r_V - r_{min}|)$$

as an expression involving four predicates, where $\mathcal{P}_2$ is subsumed by $\mathcal{P}_1$ and can be omitted.

A vertex constraint is a smooth curve in configuration space whose extent is limited by a pair of *endpoints*. In most cases, a constraint ends in the configurations where the vertex V touches one of the vertices at the ends of the edge E. At such configurations, which we call *(vertex) touchpoints*, the constraint is connected to others which also end at the touchpoint, forming a closed boundary. This is shown in Figure 14.

INSERT FIGURE 14 ABOUT HERE

When a touchpoint configuration does not fulfill the criteria for a possible configuration of touch, the constraints end instead in extremal configurations where the vertex points either straight towards or away from the center of rotation of the other object. The two possible extremal configurations are shown in Figure 15.

INSERT FIGURE 15 ABOUT HERE

In an extremal configuration, the point of touch falls on the line between the centers of rotation of the objects: they are thus identical for original and mirror image of a constraint, and an extremal endpoint connects a constraint to its mirror image.

An unachievable touchpoint between vertices of radius r_{v1} and r_{v2} is replaced:

- by an extremal endpoint of type 1 whenever d $> r_{v1} + r_{v2}$
- by an extremal endpoint of type 2 whenever d $< |r_{v1} - r_{v2}|$

Because edges are subdivided so that the radius is a unique parameterization, a vertex at the end of an edge has either minimum or maximum radius of the edge, so that these predicates are already part of the set of predicates P_1 through P_4 above.

Figure 14: A vertex touchpoint is a common endpoint of up to four different constraints. Left, the types of touch corresponding to the constraints, right, the image in configuration space.

Figure 15: Constraints ending in extremal configurations of type 1 (left) and type 2 (right).

We have thus seen how the set of active constraints and their connections is determined using a fixed set of metric predicates. As an optimization of the computation, it is also useful to eliminate those constraints that would be generated by a *concave* vertex or arc, as any configuration on them necessarily entails overlap of the objects and is thus illegal. However, this results in an incomplete set of metric predicates, which then have to be recomputed whenever the underlying convexity assumptions changes.

4.4 Local properties

As shown in Figure 14, a touchpoint is a common endpoint of up to four constraints. However, only two of the four constraints can actually be part of the boundary of free space, the other two are *subsumed*. The subsuming pair of constraints is determined by comparing the directions of the constraint curves at the touchpoint. The tangent vector of a constraint at the touchpoint is obtained by differentiating the constraint equation and instantiating it at the touchpoint coordinates.

A constraint $\mathcal{C}_1$ subsumes another constraint $\mathcal{C}_2$ if the direction vector $dir(\mathcal{C}_1)$ points into the free-space side of $\mathcal{C}_2$, as shown in Figure 16.

INSERT FIGURE 16 ABOUT HERE

The predicate that expresses this is that the scalar product of the direction vector of $\mathcal{C}_1$ with the leftward normal of the direction vector of $\mathcal{C}_2$ must be positive. The four constraints at a

Figure 16: Constraint $\mathcal{C}_1$ subsumes $\mathcal{C}_2$ when its direction vector forms a positive scalar product with the leftward normal of $\mathcal{C}_2$.

touchpoint define 6 different pairs of constraints and thus 6 predicates for relative subsumptions which select the two constraints at the boundary of free space.

At their endpoints, boundary constraints connect smoothly to adjacent vertex constraints. Describing the predicates which govern the selection of the constraint to which it is connected is beyond the scope of this discussion. The reader is referred to [FALT87b].

Another local property of constraints is their qualitative direction in configuration space. As described in [FALT90]), this qualitative direction determines the set of qualitative inference rules that hold in the corresponding configuration of touch. At a touchpoint, the qualitative direction is determined by instantiating the tangent vector and keeping the signs of its components, as described above. However, the qualitative direction of a constraint can change at configurations of $nonmonotonicity$, as shown in Figure 17.

INSERT FIGURE 17 ABOUT HERE

It turns out that the nonmonotonicity shown on the left in Figure 17 arises when the point of touch falls on a minimum of the radius, and is thus already handled by the breakup imposed by the boundary parameterization. The condition for the second kind of nonmonotonicity can be expressed by the radius of the point of touch:

$$r_{non-mon} = \sqrt{d^2 - r_v^2 \pm 2r_v r_{min}}$$

where r_{min} is the distance of the edge from the center of rotation, r_v is the radius of the vertex, and the sign is negative if the center of rotation lies to the inside of the edge, positive if it

Figure 17: The two forms of nonmonotonicity. On both the left and the right, cases a) and b) are examples of the same kind of contact, but require different inference rules.

lies on the outside of the edge. An analogous expression exists for edges which are circular arcs ([FALT87b]). Because the segments have been broken up to guarantee a unique parameterization by the radius, there can be at most one such nonmonotonicity on a constraint segment.

Because of the fundamental importance of the qualitative direction for kinematic behavior, constraint segments must be subdivided at points of nonmonotonicity. The predicates that indicate whether or not a subdivision is required are based on the condition that the edge contains a point whose radius value satisfies the above criteria, that is:

$$(r_{min} < r_{non-mon}) \wedge (r_{non-mon} < r_{max})$$

When a nonmonotonicity exists, this indicates that the constraint consists of two segments whose qualitative directions differ in the sign of the coordinate corresponding to the object containing the vertex generating the constraint.

Note again that all the local properties described in this section - subsumptions at touchpoints, connections to adjacent constraints and qualitative directions - are determined by a fixed set of metric predicates.

4.5 Free Subsumptions

Besides the connections at their endpoints, which can be analyzed locally, constraints can *subsume* each other anywhere along their valid segments. Such free subsumptions occur whenever there are two distinct points of contact between the objects. For example, the configuration of the escapement shown in Figure 18 is a subsumption between the points of touch at the left and right ends of the pallet.

INSERT FIGURE 18 ABOUT HERE

Because of their non-local nature, subsumptions must be hypothesized for any pair of constraints, a potentially very large number. This makes finding all free subsumptions that occur in a kinematic pair the part of kinematic analysis which generates by far the largest number of metric predicates. In practical mechanisms, free subsumptions occur mostly as design flaws, with the notable exception of gearwheels. This reflects the fact that people have difficulty with analyzing free subsumptions also. However, it may well be possible that mechanisms which use free subsumptions could be better than traditional devices at certain tasks.

In configuration space, a free subsumption is an intersection of two constraint curves. A single intersection between two non-closed curves appears or disappears exactly when one of the endpoints falls on the other curve, as shown on the left in Figure 19.

INSERT FIGURE 19 ABOUT HERE

This condition can be formulated as a predicate by plugging the constraint endpoint coordinates of one constraint into the equation of the other, resulting in a closed-form expression which becomes zero whenever a subsumption appears or disappears. A *pair* of free subsumptions can also arise when there is a pair of intersections between constraints. In this case, one of the constraints is *locally* subsumed by the other, without global influence on the kinematics. Using

Figure 18: A (nonfunctional) escapement. The lever can touch the wheel at both its left and its right end.

Figure 19: A single intersection between constraints appears when one of the endpoints falls on the other constraint (left). Double intersections can also appear elsewhere on the curve.

algebraic decision methods, it is possible to formulate predicates that test for the existence and number of intersections of algebraic curves (see [BKR85]). In our current implementation, double subsumptions are ignored, and this resulted in no errors in the more than 20 examples we analyzed.

The problem with the predicates generated by endpoint tests is that they are not sufficient to determine the place vocabulary. The constraint equations define closed curves in configuration space, of which the constraints themselves are only subsegments. Testing whether an endpoint falls on a constraint curve does not distinguish between the valid and invalid segments of a constraint curve. The predicates generated by the endpoints are thus not sufficient to reliably determine all existing subsumptions, as illustrated in Figure 20.

INSERT FIGURE 20 ABOUT HERE

With respect to the generation of metric predicates, this phenomenon means that the predicates generate too many distinctions. However, this does not cause any difficulty for the enumeration analysis, because the presence of a actual distinctions is verified by computing the corresponding place vocabularies.

For computing an actual place vocabulary, and for dependency-directed reasoning based on it, it is necessary to construct a different set of predicates, called *eval predicates*. The eval predicates we construct have the advantage that they reliably predict all single constraint intersections, but the disadvantage is that their formulation dependends on actual parameter values and they thus can not be used as a basis for enumeration analysis (which, however, is done correctly using

Figure 20: Testing the relative location of constraint endpoints correctly predicts the intersection between A and C, as their endpoints lie on opposite sides of each other's algebraic curve. This is not the case for the intersection between B and C, where for each constraint both endpoints lie on the same side of the other's algebraic curve.

endpoint tests).

We can turn the endpoint tests into reliable intersection tests if they can be restricted to those parts of the curves which are actually valid constraints. Such a restriction can be achieved by constructing subdivisions as follows. We define the *bounding rectangle* of a constraint segment to be the rectangle parallel to the coordinate axes whose opposite corners are the endpoints of the constraint segment. We iteratively break each monotone constraint segment in half until the configuration space area covered by each of the bounding rectangles contains only valid portions of the constraint curve, as shown in Figure 21. For a pair of constraints, all intersections between valid segments must fall within intersections of bounding rectangles of its subsegments. Let I be the rectangle of intersection between the bounding rectangles of two different constraint segments. Using the points where the constraints enter I as endpoints for the intersection test, the test is now reliable as no intersections with invalid portions of the curves can occur within I. The required endpoint can be computed iteratively using the constraint parameterization, as discussed in [FALT87b]. While these predicates can be used for dependency-directed perturbation analysis, note that their formulation depends strongly on the actual parameter values, making them meaningless for reasoning about large parameter variations as is done in enumeration analysis.

For constraints which have several free subsumption intersections, it is necessary to know in which order they occur along the constraint. For cases where these intersections occur very close to each other, so that no distinction by the bounding rectangles is made, this ordering is determined by locating points between the different intersections, as described in [FALT87b]. Gener-

Figure 21: A constraint broken up into monotone subsegments whose bounding rectangles contain only valid portions (left), and testing for subsumptions between a pair of constraints using the subsegments (right).

ating metric predicates for this case requires using algebraic decision methods (see [FALT87b]) and has not been implemented.

With respect to the generation of landmark values, it is important to note that our analysis is incomplete in two respects: it does not take into account changes in the ordering of subsumption intersections, and it also ignores double intersections between constraints. Both of these shortcomings could be corrected using algebraic decision methods ([BKR85, KY85]). However, these methods are very inefficient: an algebraic intersection test for a single pair of constraints would involve thousands of metric predicates. Furthermore, the actual condition for intersection involves a complicated logical combination of their values, making a straightforward inversion impossible. The shortcomings of the current methods have caused no difficulties in any of the examples they have been used on, and we have therefore not investigated the possibility of algebraic methods any further.

4.6 Determining the Output Place Vocabulary

The place vocabulary is a description of the legal parts of configuration space and their boundaries. The tests described earlier in this section define the graph of valid configuration space constraints and their intersections. Tracing the cycles of this graph determines the regions of free space and their boundaries, and thus the place vocabulary. No additional metric information is necessary for extracting the place vocabulary from the graph, as described in detail in [FALT87b]).

In order to reduce reasoning ambiguities, it is sometimes useful to impose additional subdivisions of the regions of free space ([FALT90]). These subdivisions are determined by a partial order of the coordinates of the constraint endpoints and intersections along both coordinate axes. For constraint endpoints, this ordering is given by a set of predicates comparing the relative coordinates. For free subsumption intersections, it must be determined by numerical computation, and no landmark values can be given for them except by algebraic decision methods.

5 Other Work

Mechanism kinematics is an important area of study in mechanical engineering, and many methods exist which have similar goals as the ones described in this paper. They can be grouped into two categories: classification of mechanisms according to types of behavior, and analysis of particular states to avoid undesirable properties or behavior.

A famous example of mechanism classification are Grashoff's criteria for the classification of four-bar linkages. The possible motions of a four-bar linkage, shown in Figure 22, can be classified according to whether a full rotation is allowed (crank) or not (rocker) for links A and C. This results in the categories

- double crank, when link D is shorter than all others
- double rocker, when link B is the shortest
- crank rockers, when link A or C is the shortest

Figure 22: A four-bar linkage

An additional condition is given by difference between the sum of the lengths of the shortest and the longest links and the sum of the other two links. If this difference is greater than zero, only double rocker mechanisms are possible. If it is equal to zero, a linkage with a singular state (two possible behaviors) will result ([HADE64]).

Similar to the approach described in this paper, the conditions for the different classes are expressed as conditions on their dimensions, and correspond to a particular division of metric space. However, an important difference is that the criteria have been defined by *manual* analysis of a particular device topology, the four-bar linkage. They do not automatically generalize to other mechanisms of the same class, as is the case for causal inversion methods.

The second category of mechanical problems for which similar results exist is the detection (and avoidance) or singular or undesirable states in the behavior of a mechanism. As an example, consider the singular state of a *slider-crank* mechanism shown in Figure 23. Mechanical engineers know that such a state exists whenever link a is longer than link b. In general, singular configurations exist whenever the mechanism's system of kinematic constraint equations has a singularity ([HADE64]). This criterion can be expressed as a set of conditions on the metric dimensions, and form the basis for a decomposition of metric space analogous to that constructed in metric space. Again, the methods depend heavily on a manually constructed model of the device and can not be automated in a straightforward way.

Our techniques of causal inversion could in principle be used to automate both kinds of analysis. However, the theory of qualitative kinematics which is used as a basis for our investigation

Figure 23: A slider-crank mechanism, used to translate rotation of part A into translation of part B. The state shown in singular in that a force on A will not allow it to move.

is currently restricted to fixed-axis mechanisms, and excludes the domain of linkages in which these criteria have been formulated.

As a computational analysis, the work of Kramer and Barrow ([KRBA90]) on TLA has some similarities with the approach in this paper. They show how suitable dimensions of linkages can be chosen by continuous optimization of an initial guess, selected from a large catalog of representative examples. The goal-directed generation of modification operators performed by perturbation analysis is implicit in the optimization procedure, which is generated automatically. The initial catalogue is somewhat similar to the global analysis embodied in enumeration analysis. TLA is similar to causal inversion methods in that it uses an automatic first-principles analysis, but adresses a very different domain (linkages instead of higher kinematic pairs), and does not make behavior conditions or modification operators explicit.

6 Conclusions

In this paper, we have presented methods to solve the problem of symbolic reasoning about the influence of the metric dimensions of object shapes on kinematic interactions. Because of the non-linear interdependence of the metric parameters, such reasoning is not possible using independent qualitative representations commonly used in qualitative physics. Instead, we have shown how the influence of all metric dimensions can be restricted to a set of predicates. The set of predicates defines a combined qualitative representation of all metric parameters, out of which individual qualitative representations can be extracted for reasoning.

This technique is an extension of the available representation formalisms of qualitative physics and could be used for modeling other physical systems where the simultaneous influence of several parameters is difficult to decouple. The only condition that has to be satisfied is that the functional analysis is expressed in a finite symbolic structure, and that this structure can be derived using a fixed set of metric predicates. This condition is not trivially satisfied for any system. For example, it is typically not satisfied for unbounded dynamic parameters. Consider the example in Figure 24, in which the interesting question is whether the billard ball will make it into the pocket or not.

INSERT FIGURE 24 ABOUT HERE

Depending on the number of traversals of the board that the ball can make before coming to rest, there exist a variable number of possible intervals of initial directions of motion which will make

Figure 24: A prediction problem in billard. The goal is to determine which combinations of initial velocities will make the ball drop into the pocket. Depending on the number of traversals, there are an infinite number of disjoint conditions.

it enter the pocket. This results in an unbounded set of conditions which can not be expressed by a fixed set of predicates. At this point, we do not have a characterization of the class of problems to which the methods of this paper could be applied.

Besides its utility for reasoning, explicit calculation of metric predicates also explains why people can so easily state the conditions on parameter values under which certain behaviors occur. This capability is difficult to explain in a model where metric information is used in a less structured way. The metric predicates also allow cases where metric information is insufficient to be modeled as ambiguities, another interesting aspect of human behavior that is poorly explained by numerical analysis methods.

We have shown how metric predicates make it possible to use dependency-directed reasoning techniques for reasoning about the relationship between metric dimensions and kinematic function. However, the techniques are (and cannot be) as powerful as one would like. Because unrelated object parts can interfere with each other's kinematic interaction, operators based on local dependencies can never be guaranteed to succeed. Furthermore, they can only change or eliminate functions that already exist - no dependencies are provided for elements which are not part of the place vocabulary of the current device[9].

A limitation of our methods is that they only allow reasoning about a single parameter at a time. This makes the system incomplete, as situations which require the simultaneous mod-

[9]With some exceptions: it is possible to give conditions for making inactive constraints active, and it is possible to label subsumed constraints with the subsuming ones and their conditions.

ification of several parameters are beyond its capabilities. Although there do exist algebraic algorithms for analyzing the possibilities of simultaneous variation of several parameters, they are so hopelessly inefficient that they do not seem practical even on extremely powerful computers.

Finally, all of the kinematic analysis methods presented in this paper are restricted to interactions of pairs of two-dimensional objects, each of which has only one degree of freedom. We are currently working on new methods which have the promise to be generalizable to higher dimensions and more degrees of freedom.

6.1 Acknowledgements

I would like to thank Ken Forbus, Tom Galloway and Paul Nielsen for comments on this research. Part of this work was carried out while the author was at the University of Illinois, where he was supported by an IBM graduate fellowship and ONR under contract No. N-00014-85-K-0225.

References

[ADA87] **S. Addanki, S. Penburthy, E. Davis**: "Envisioning as Problem Solving: An Application of Prompt," IBM Research Report, 1987.

[BKR85] **Ben-Or, D. Kozen, J. Reif**: "The Complexity of Elementary Algebra and Geometry," Journal of Computer and System Sciences **32**, 1986, pp. 251-264.

[DKL75] **Johann DeKleer**: "Qualitative and Quantitative Knowledge in Classical Mechanics," MIT AI TR-352, 1975.

[HADE64] **R. S. Hartenberg, J. Denavit**: "Kinematic Synthesis of Linkages," McGraw-Hill, 1964

[DOY79] **Jon Doyle**: "A Truth Maintenance System," Artificial Intelligence **12** (1979).

[FALT87a] **Boi Faltings**: "Qualitative Kinematics in Mechanisms," Proceedings of IJ-CAI 87, Milan, Italy, August 1987.

[FALT87b] **Boi Faltings**: "The Place Vocabulary Theory of Qualitative Kinematics in Mechanisms," University of Illinois Technical Report UIUCDCS-R-87-1360, July 1987.

[FALT90] **Boi Faltings**: "Qualitative Kinematics in Mechanisms," *Artificial Intell-gence* **44**(1), June 1990.

[FNF91] **Ken Forbus, Paul Nielsen, Boi Faltings**: "Qualitative spatial reasoning: The CLOCK project," *Artificial Intelligence* **51**(1-3), October 1991

[FOR80] **Ken Forbus**: "Spatial and Qualitative Aspects of Reasoning about Motion," Proceedings of the National Conference on AI, Stanford, August 1980.

[FOR84] **Ken Forbus**: "Qualitative Process Theory," Artificial Intelligence **24**, 1984.

[HAY79] **B. Hayes**: "The Naive Physics Manifesto," in: Expert Systems in the Micro-Electronic Age, D. Michie (ed.), Edinburgh University Press, May 1979.

[KRBA90] **Glenn Kramer, Harry Barrow**: "An Intelligent Mixed Symbolic/Numerical System for the Design of Mechanical Linkages," in: T. Tomiyama (ed.), Intelligent CAD 2, North Holland, 1990

[KY85] **D. Kozen, C. K. Yap**: "Algebraic Cell Decomposition in NC," extended abstract, Proceedings of the 26th FOCS, Oct. 1985.

[LMT83] **T. Lozano-Perez, M. Mason, R. Taylor**: "Automatic Synthesis of Fine-Motion Strategies for Robots," MIT AI Memo 759, 1983.

[LPW79] **T. Lozano-Perez, M. Wesley**: "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," Comm. of the ACM, Vol. 22 (1979), pp. 560-570.

[MARR81] **David Marr**: "Vision," Freeman and Company, 1981.

[PEN87] **J. Scott Penburthy**: "Incremental Analysis in the Graph of Models: A First Step Towards Analysis in the Plumbers World," S.M. Thesis, MIT, 1987.

[SHS83] **J. T. Schwartz, M. Sharir**: "On the Piano Movers Problem. II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds," Advances in Applied Mathematics,**4** (1983), pp. 298-351.

[ULL84] **S. Ullman**: "Visual Routines," Visual Cognition, 1984.

[WEL88] **D. Weld**: "Theories of Comparative Analysis," MIT AI Lab. Technical Report 1035, 1988.