Assembling a Device

Jean-Luc Dormoy
Electricité de France Research Center
Clamart, France

Olivier Raiman
I.B.M. Scientific Center
Paris, France

LAFORIA
Pierre & Marie Curie Paris University

Abstract

We present here a new way of reasoning on a device based on structure, we call assembling a device. It consists of a symbolic combination of local qualitative constraints (namely confluences) leading to more global relations. Some reference variables are selected according to the task to be performed (simulation, observation, postdiction,...). The assembling step produces a set of equations expressing directly "internal" quantities as functions of the reference quantities. We call such a set a task-oriented assemblage. Then, determining the non ambiguous variables for a particular assignment of the reference quantities turns out to be straightforward. We can thus expect to perform qualitative reasoning on large systems.
The assembling tool is a new rule, we call the qualitative resolution rule. It has agreable properties: (1) interpretation: each application can be interpreted as joining local descriptions to more global ones; (2) completeness: an assemblage provides all the non ambiguous variables for any assignment of reference variables.

1 Introduction

Qualitative reasoning about a physical device is an attempt to make a computer focus on the device properties in the same way an engineer does. A typical problem is to capture key features of the device behavior. This has been the main concern for people working in the Qualitative Physics area.
This work shows how a computer program can deduce global properties specific to a device by combining local physical laws. Essentially we attack the problem by defining a new task, which is not based on causality, but on the idea of assembling the components of the device.
Technically speaking, this task is performed by a single rule, we call the qualitative resolution rule. First we show on some simple and motivating examples how the resolution rule, by assembling the device, produces global laws. Thus, performing simulation or other tasks, such as observation, turns out to be straightforward. This enables us to produce very efficient task-oriented programs, even for large-scale plants.
Then we describe precisely how the resolution rule must be applied. This leads us to prove some basic properties of the signs algebra.
Then we state a completeness result: all the non ambiguous physical quantities can be drawn from global laws produced by the resolution rule. Such a set of global laws is called an assemblage.
In practical terms, we specify the form of the global laws composing an assemblage. This enables us to stop firing the qualitative resolution rule as soon as it has provided an assemblage.
We conclude by a comparison to De Kleer's and Brown's work.

2 Assembling some devices

2.1 Is the sum of two pipes a pipe?

Consider a very simple example, a qualitative model for two connected pipes (Fig. 1).

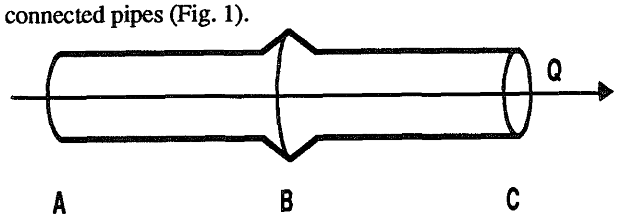


Figure1: Two connected pipes

For each pipe, there is a confluence describing the link between the sign of the pressure at the different ends of the pipe and the flow Q. The confluence (1) resp. (2) for pipe 1 and pipe 2 are the following:

$$[dP_A]-[dP_B]-[dQ]\approx 0 \qquad (1)$$
$$[dP_B]-[dP_C]-[dQ]\approx 0 \qquad (2)$$

This model describes separately the different parts of the physical device. It is obvious that two connected pipes behave like a single pipe. This means that the following confluence must hold:

$$[dPA]-[dPC]-[dQ]\approx 0 \qquad (3)$$

A system performing qualitative reasoning should be able to deduce (3) from (1) and (2).

2.2 The qualitative resolution rule

Deducing confluence (3) from confluences (1) and (2) requires eliminating variable $[dP_B]$. The trouble is that gaussian elimination is not correct in general for confluences. The following rule states under what conditions such an elimination can be performed:

Qualitative Resolution Rule: Let x, y, z, a, b be qualitative quantities such that

$$x + y \approx a$$

and $-x + z \approx b$

If x is different from ?, then

$$y + z \approx a + b$$

Detailed explanation, proof and related properties are given below.

Take $x = [dP_B]$, $y = -[dP_C] - [dQ]$, $z = [dP_A] - [dQ]$ and a and $b = 0$. As $[dP_B]$ is a physical quantity, its range is $\{0, +, -\}$. Hence the conclusion can be drawn:

$$[dP_A] - [dP_C] - [dQ] \approx 0 \qquad (3)$$

Moreover, the qualitative resolution rule provides another confluence by "subtracting" confluences (1) and (2) and "eliminating" $[dQ]$:

$$[dP_A] - [dP_B] + [dP_C] \approx 0 \qquad (4)$$

Initial confluences (1) and (2) describe links between the physical variables involved in the elementary components pipe 1 and pipe 2. The inferred confluences (3) and (4) describe the consequences of connecting the two pipes: they are specific properties of the composite device. The qualitative resolution rule discovers global relations starting from local ones.

2.3 Consequences for a simulation task

A classical task which a qualitative reasoner should be able to perform is simulation, that is predicting the behavior for a given input. For example, we would like to perform a simulation under the assumptions $[dP_A] = +$ and $[dP_C] = 0$. We can use two obvious rules, we call in this paper propagation rules:

PR1: If the value of a variable x is known, then substitute x by its value in all the confluences mentioning x.

PR2: If an equation mentions exactly one variable, then deduce its value.

Consider the inferred confluences (3) and (4). It is obvious using the two propagation rules that $[dP_B] = +$ and $[dQ] = +$. The reason for being able to draw these conclusions is that confluences (3) and (4) are global behavioral descriptions of the device as a whole, linking explicitly the internal variables $[dP_B]$ and $[dQ]$ to the input $[dP_A]$ and $[dP_C]$:

$$[dP_B] \approx [dP_A] + [dP_C] \qquad (A1)$$

$$[dQ] \approx [dP_A] - [dP_C] \qquad (A2)$$

By assembling the two pipes and providing the global behavioral relations (A1) and (A2), we have reduced (in this case - based on the quasi-static assumption) any simulation task to simple propagation. For instance, it would have been as easy to predict the behavior of the device starting from other input values:

$[dP_A] = 0$, $[dP_C] = +$ ==> $[dP_B] = +$, $[dQ] = -$

$[dP_A] = +$, $[dP_C] = +$ ==>

$[dP_B] = +$, $[dQ]$ remains unknown

The last case is important. It is possible to compute the value of $[dP_B]$, but the propagation rules lead to an ambiguous value for $[dQ]$: $[dQ] \approx [+] - [+]$. It is often the case that some quantities are determinate as others remain ambiguous. The method introduced here is not responsible of the ambiguity of $[dQ]$, but the "qualitativeness" of the model is. On the other hand, $[dP_B]$ is not ambiguous in the model, and its value is inferred.

Now, forget for a while that confluences (3) and (4) can be inferred and apply propagation directly to the initial confluences (1) and (2). This gives: $-[dP_B] - [dQ] \approx -$ (1) and $[dP_B] - [dQ] \approx 0$ (2). No other information can be gotten, except by using some kind of indirect proof. By itself, propagation is incomplete. This example shows intuitively the advantage and the meaning of the resolution rule:

- it seems to reduce simulation to simple propagation, while propagation by itself is incomplete.
- at the same time, it assembles the parts of the device and provides global properties specific of the compound device.

We will now give a deeper insight into the nature of what is assembling a device.

2.4 Assembling the device for simulation

Consider a general device with input $i_1, \ldots, i_p$, internal variables $v_1, \ldots, v_p$ and a qualitative model based on confluences involving the qualitative derivatives of these quantities. Suppose we want to build a system which can answer quickly any simulation-like question : "How does the device react to input $[di_1] = a_1, \ldots, [di_p] = a_p$?"

This can be done in two steps (Fig. 2):

- Assembling the device, that is obtaining from the initial qualitative model global relations, for instance relations expressing directly the internal variables as functions of the input: $[dv_j] \approx f_j([di_1], \ldots, [di_p])$, $1 \le j \le n$ These relations will hold whatever values are assigned to the input.
- Then propagating input values into these global laws.

The second step relies only on the two basic propagation rules. In our first example, the first step is achieved using the resolution rule.

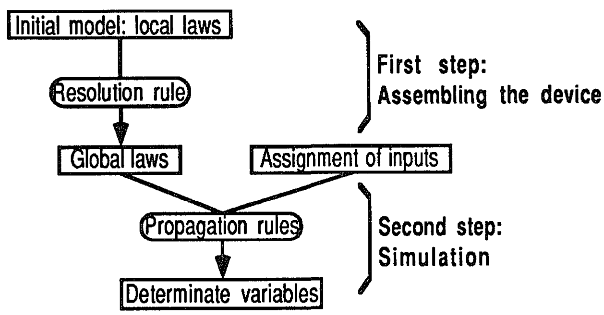


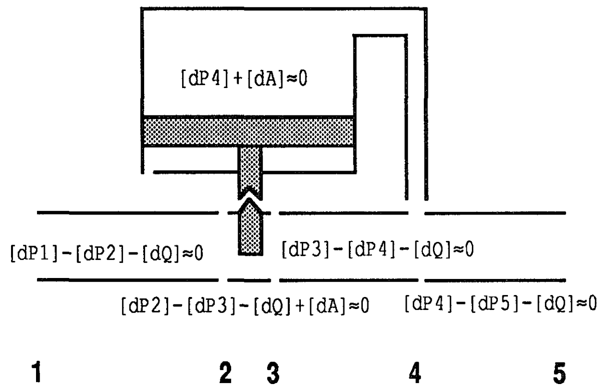
Figure 2: Simulation in two steps

Solving confluences happens to be an NP-complete problem

[Dormoy, 1987]. At first sight, if there are k simulations to be performed, then we can expect to be confronted k times with a (probably) exponential problem. Thus, splitting simulation into these two steps is fundamental. The first step is NP-complete too, but it is done once and for all, The second step will be performed k times, but it is known to be polynomial (in the worst case, $O(nxp)$).
The first step can be viewed as compiling the device for simulation, and so avoids "re-interpreting" the initial set of confluences for each new simulation. The second step can be coded as a very simple and efficient program. This program is specific to the device, but this is why it is efficient. We may thus expect to perform on-line simulations on large-scale plants having multiple input variables.

2.5 The pressure regulator revisited

Consider a second example, the well known pressure regulator. The model used here (Fig. 3) is slightly different from De Kleer's and Brown's [1984]:



$[dP1]-[dP2]-[dQ]\approx 0$ (1)
$[dP2]-[dP3]-[dQ]+[dA]\approx 0$ (2)
$[dP3]-[dP4]-[dQ]\approx 0$ (3)
$[dP4]-[dP5]-[dQ]\approx 0$ (4)
$[dP4]+[dA]\approx 0$ (5)

Figure 3: The pressure regulator and its model

P_1 and P_5 are the input variables, P_2, P_3, P_4, Q and A are the internal variables. Assembling the pressure regulator for simulation using the resolution rule is possible. For instance, we can get the relation involving $[dP_2]$:

$[dP_2]\approx[dP_1]+[dP_5]$ (A1)

in four steps (Fig. 4):

$[dP2]-[dP3]-[dQ]-[dP4]\approx 0$ (6)=(2)-(5)
$[dP_2]-[dP_4]-[dQ]\approx 0$ (7)=(6)+(3)
$[dP_2]-[dP_5]-[dQ]\approx 0$ (8)=(7)+(4)
$[dP_1]-[dP_2]+[dP5]\approx 0$ (9)=(1)-(8)

In the same way the resolution rule provides the following global laws ($[dP_3]$ will be given later on):

$[dP_4]\approx[dP_1]+[dP_5]$ (A2)
$[dQ]\approx[dP_1]-[dP_5]$ (A3)

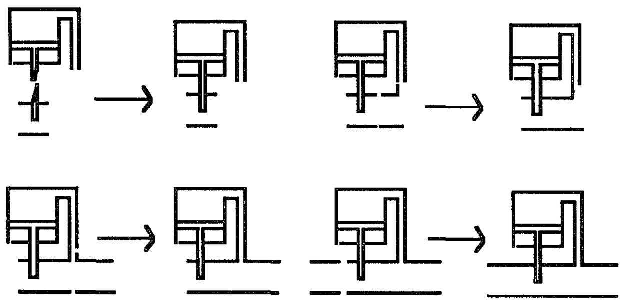


Figure 4: Assembling the pressure regulator for simulation

$[dA]\approx-[dP_1]-[dP_5]$ (A4)

As in the example of the two pipes, simulation is now reduced to propagation.

2.6 Assembling the device for postdiction

As expected, the resolution rule assembles the device for simulation. But this is not the only point. This example highlights other tasks that can be performed using resolution. Imagine that we cannot directly observe the input, but that we can measure the evolutions of [dA] and [dQ]. We are no longer interested in simulation, but in postdiction : "what input has caused the fact that [dA]=a and [dQ]=q ?" Formally, this problem is very similar to simulation: solving it only requires expressing the other variables as functions of [dA] and [dQ]. The general task of assembling a device can still be performed, whatever set of reference variables is selected. The global laws of the pressure regulator for reference variables [dA] and [dQ] are:

$[dP_1]\approx[dQ]-[dA]$
$[dP_2]\approx[dQ]-[dA]$
$[dP_3]\approx[dQ]-[dA]$
$[dP_4]\approx-[dA]$
$[dP_5]\approx-[dQ]-[dA]$

We can thus expect to observe a device with the same advantages as for simulation.

3 Scanning the qualitative resolution rule

Before discussing about what the resolution rule can indeed produce, we need to pause and see exactly what it is. We are starting by the proof, for it may avoid possible confusion.

3.1 Proof

The qualitative resolution rule can be stated in several ways. We gave the shortest and the most general one:

Qualitative Resolution Rule: Let x, y, z, a, b be qualitative quantities such that

$x + y\approx a$

and $-x + z\approx b$

If x is different from ?, then

$y + z\approx a + b$

Before proving the rule, we need two basic qualitative calculus properties:

Quasi-transitivity of qualitative equality:

If $a \approx b$ and $b \approx c$ and $b \neq ?$, then $a \approx c$.

Compatibility of addition and qualitative equality:

$a + b \approx c$ is equivalent to $a \approx c - b$

It is very easy to prove these properties provided that the relation $\approx$, called qualitative equality, or sign compatibility, is properly defined:

$a \approx b$ iff $a = b$ or $a = ?$ or $b = ?$

This relation is not the usual equality. Let F_1 and F_2 be two expressions, involving additions and products of physical quantities, such that $F_1 = F_2$, and E_1 and E_2 the corresponding qualitative expressions. $E_1 \approx E_2$ means that the resulting signs of the two expressions F_1 and F_2 must be compatible. Suppose we have assigned some values to the physical variables involved in both F_1 and F_2, and let s_1 and s_2 be the corresponding values of E_1 and E_2. If s_1 and s_2 are non ambiguous, i.e. are both different from ?, they must be equal; but if one of them is ? (the sum of a + and a –), the underlying real expression may have any sign; hence, it may be compatible with any other sign.

All this is obvious and well-known. The point is that as soon as we have formally defined the set $S = \{+, 0, -, ?\}$, its addition and product, as well as the qualitative equality, we may work within this structure and prove things while forgetting the initial motivation. For instance, the proof of the first property is:

If $a=?$ or $c=?$, then obviously $a \approx c$.

Otherwise $a=b=c$. ◊

No matter what a, b or c are.

The second property can be proved by case analysis.

We can now give the proof of the rule:

Let x, y, z, a and b be like in the first statement of the rule:

$x + y \approx a$

$-x + z \approx b$

We get, by applying the second property:

$z - b \approx x$ and $x \approx a - y$

The assumption $x \neq ?$ allows us to apply the first property:

$z - b \approx a - y$

which can be rewritten using again the second property:

$y + z \approx a + b$ ◊

This proof needs a comment. Probably the most expressive way to state the resolution rule is:

A variable may be eliminated by adding or subtracting two confluences, provided that no other variable is eliminated at the same time.

But this may lead to confusion: it could be thought that we may add or subtract two confluences, and then eliminate a variable by the "elimination rule" x-x --> 0. This is clearly a wrong statement, since x-x is hardly ever 0, unless x itself is 0. But the resolution rule states that one can proceed as if this were true, and provided that one applies the "elimination rule" only once. This is clearly not the way the rule is proved.

3.2 Using the resolution rule in the right way

We have shown in the examples above how one succeeds in firing the resolution rule. We show here how one can fail.

In practical terms, the relations $x+y \approx a$ and $-x+z \approx b$ stand for some confluences, and x is a variable involved in both. The hypothesis *if x is different from ?* is thus always verified, since x stands for a qualitative derivative of a physical quantity. In order to obtain the exact pattern of the rule, the second confluence may be multiplied by – if necessary.

a and b are the respective right-hand sides of the confluences (until now 0). y and z are the remaining expressions of the respective left-hand sides after having removed x. y and z may involve a common variable. There is a problem if y and z involve a variable t with opposite coefficients: when adding y+z, we get t-t, which cannot be simplified (it is not correct to substitute 0 for t-t, cf. previous remark). Otherwise, we get t+t or -t-t, which can be simplified according to the rule t+t=t. All this is better illustrated by the following examples (borrowed from the pressure regulator).

♦ Consider first the two confluences:

$[dP_2]-[dP_3]-[dQ]+[dA] \approx 0$ (2)

$[dP_4]+[dA] \approx 0$ (5)

They have a single variable in common, x=[dA]. We must consider the opposite of confluence (5). y and z have no variable in common: $y=[dP_2]-[dP_3]-[dQ]$, $z=-[dP_4]$. The resulting confluence is:

$[dP_2]-[dP_3]-[dQ]-[dP_4] \approx 0$

(6)=(2)-(5)

♦ Let's try now to combine this confluence and confluence:

$[dP_3]-[dP_4]-[dQ] \approx 0$ (3)

There are three possible choices for x: $[dP_3]$, $[dP_4]$ and [dQ]. Let's try $[dP_3]$ first.

We have $y=-[dP_4]-[dQ]$

and $z=[dP_2]-[dP_4]-[d_Q]$.

y and z have two variables in common, and we are in the case t+t. Hence we get:

$[dP_2]-[dP_4]-[dQ] \approx 0$ (7)=(6)+(3)

♦ Let's now try $x=[dP_4]$ starting with the same confluences (3) and (6) (choosing x=[dQ] would lead to a similar conclusion). We obtain $y=[dP_2]-[dP_3]-[dQ]$ and $z=[dP_3]+[dQ]$. We are in the case t-t. The relation $y+z \approx 0$ is of no practical use. Such applications of the resolution rule must be avoided.

For functional purposes, the resolution rule must be stated as follows:

Let $x+E_1 \approx a$ and $-x+E_2 \approx b$ be two confluences, where x is a variable and E_1 and E_2 have no variable with opposite coefficients in common. Then $E_3 \approx a+b$ is a valid confluence, where E_3 is the same expression as E_1+E_2, but with no repeated variable.

3.3 Why *resolution*?

We had called the qualitative resolution rule the qualitative Gauss rule, because of its similarity with gaussian elimination. But another analogy seems stronger. The qualitative resolution rule and the Resolution Rule in logics (weakened here to the propositional calculus) have a similar aspect:

Let X, Y, Z be propositional variables (and x, y, z their boolean equivalents) such that

$$X \vee Y \qquad (x + y = 1)$$

and $\neg X \vee Z \qquad (-x + z = 1)$

Then

$$Y \vee Z \qquad (y + z = 1)$$

Moreover, the two resolution rules have completeness properties (see below).

It must be mentioned that there is a third resolution rule, valid in a model dealing with orders of magnitude (which embeds the standard signs model). We have proved no completeness result within this framework, but we guess that there is one.

We are thus facing a situation with three similar rules and two completeness results (probably three) in models of increasing complexity: there is something fishy going on. But we have not caught it yet.

4 Completeness of qualitative resolution

4.1 Power of the resolution rule

We have shown in the examples the advantage of performing the task we have called assembling a device: the resolution rule provides relations, from which the basic propagation rules are powerful enough tools to determine the non ambiguous variables and their values. Efficient programs could be designed in this way. But are we sure that this works in all cases ?

This is a completeness problem. For instance, in the two pipes case, the values for $[dP_B]$ and $[dQ]$, when not ambiguous, are imposed by the model, not by a particular method. The challenge, when proposing an effective method, is to know whether it can reach all that is embodied in the model. This is not true for the propagation rules. But we saw that these rules could deduce all the non ambiguous values from the global laws produced by resolution whatever the assignments of reference variables were. We suspect that the resolution rule is complete in this way.

4.2 Assemblages

Which kind of global laws the propagation step needs depends on the task to be achieved, i.e. on the choice of reference variables. Suppose we have selected one. Then the resolution rule is requested to discover an assemblage: that is, a set of global laws from which the propagation rules deduce all the non ambiguous variables and their values for any assignment of the reference variables. More formally, an assemblage can be defined as follows:

Let C be a set of confluences, w_i be selected reference variables and v_j the remaining ones. A set of global laws A is called an assemblage for the reference variables w_i iff for each assignment of the reference variables $w_i = a_i$, as soon as the model C imposes the value b_j to the internal variable v_j, then the basic propagation rules can deduce $v_j = b_j$ from the assemblage.

The completeness problem comes down to obtaining assemblages for each possible choice of reference variables.

4.3 Partially proved

Indeed, though we think that it is true in any case, we have proved the completeness only in the square case, i.e. when the number of confluences and of internal variables are equal. The proof is difficult to show: it requires introducing the notions of qualitative determinant, qualitative rank, maximal matrices with full rank,... Its total length exceeds twenty pages, and therefore will not be given here (it can be found in [Dormoy, 1987]).

Incidently, this completeness result also applies when the reference set is empty. This means for instance that, if we are performing a simulation for some particular input perturbations, then the resolution rule can find out all the non ambiguous variables from the initial set of confluences as performing a simulation for some particular input perturbations, then the resolution rule can find out all the non ambiguous variables from the initial set of confluences as well. But the advantages of the assembly step would be lost if the resolution rule were to be used in this way.

4.4 The general resolution rule is needed for completeness

Unexpectedly, we discovered after having written down the completeness proof that this work was not the first attempt to seek an effective and complete method for the unicity problem in confluences. In the field of economics, Ritschard proposed a more constrained form of the resolution rule, but leading to a more informative conclusion (the divergences from the resolution rule are underlined) [Ritschard, 1983]:

Let $x+E_1 \approx a$ (C_1) and $-x+E_2 \approx b$ (C_2) be two confluences, where x is a variable and E_1 and E_2 have no variable with opposite coefficients in common. Assume that all the variables involved in E_2 are involved in E_1 as well. Then $E_3 \approx a+b$ (3) is a valid confluence, where E_3 is the same expression as E_1+E_2, but with no repeated variable. Moreover, if $a+b=b$, then substituting confluence (C_3) for confluence (C_1) provides an equivalent set of confluences.

For instance, this rule applies in the pressure regulator example to confluences (6) and (3):

$$[dP_2]-[dP_3]-[dP_4]-[dQ] \approx 0 \qquad (6)$$

$[dP_3]$ can be eliminated in confluence (6), so giving confluence (7):

$$[dP_2]-[dP_4]-[dQ] \approx 0 \qquad (7)$$

This deduction is made by the resolution rule as well, but the additional result is that confluence (6) can be discarded.

Ritschard claimed a completeness result concerning this rule. Unfortunately, his claim is wrong, as shown by the counter example:

$y+z+t \approx 0$
$x \quad -z+t \approx 0$
$x+y \quad -t \approx 0$
$x-y+z \quad \approx 0$

All the variables must be 0, but Ritschard's rule does not apply even once. It can be checked that the resolution rule works right.

5 Task-oriented confluences

The completeness result stated above theoretically proves that the resolution rule always provides an assemblage. But, in practical terms, we must describe precisely the form of the global laws composing an assemblage.
We saw in the examples that we could express an internal variable as a function of the reference ones:

$$[dv_j] \approx f_j([dw_1], \ldots, [dw_p])$$

For instance: $[dP_2] \approx [dP_1] + [dP_5]$ (A1), drawn from the global law: $[dP_1] - [dP_2] + [dP_5] \approx 0$ (9). But we saw too, when assembling the pressure regulator for simulation, that a global law mentioning $[dP_3]$ and the input $[dP_1]$ and $[dP_5]$ was missing. Completing a simulation-oriented assemblage for the pressure regulator requires extending the notion of a confluence. The following relation holds for $[dP_3]$:

$$[dP_3] \approx [dP_1] + ?[dP_5] \qquad \text{(A5)}$$

The use of ? coefficients in confluences must not make things confused. This relation means that:

- if $[dP_5]$ is different from 0, then $[dP_3]$ cannot be determined from this relation.
- if $[dP_5]=0$, then $[dP_3]=[dP_1]$ (since regular and qualitative equalities are equivalent for two qualitative quantities different from ?).

Hence, relation (A5), despite the ? coefficient, provides some information. Indeed it provides the best, since $[dP_3]$ is ambiguous as soon as $[dP_5]$ is different from 0.
In general, the way we represent physical laws must not change: confluences are suitable. But the goal to be achieved for a particular task imposes changes to their form: the reference variables must be passed to the right-hand side. The resolution rule applies in the same way, but regardless to the right-hand side. This means that one can deduce relations involving a pattern $w-w$ in their right-hand sides, where w is a reference variable. As usual, we run into ambiguity as soon as w is different from 0, but such a relation may provide some information when $w=0$. We call such relations task-oriented confluences.
The conclusion is that the resolution rule provides assemblages composed of task-oriented confluences.

6 Conclusion

The work reported here clearly is in the continuum of previous research in qualitative physics, but it relies on a different and new approach.
As said above, propagation rules are incomplete by themselves, hence a kind of indirect proof is needed. De Kleer and Brown called RAA the chronological backtracking algorithm which determines all the solutions of a set of confluences. But RAA cannot capture the way an engineer discovers how a device works. Technically speaking, causal heuristics are designed to control RAA. But they are intended to express more: they are an attempt to set within the device-centered model based on confluences the engineer's notion of causal perturbations.
Our work presents two aspects as well. From a technical point of view, the resolution rule avoids the incompleteness of propagation by discovering task-oriented assemblages. The completeness result makes this step safe. At the same time, efficient task-oriented programs are produced. But we intend more: in our opinion, assembling a device captures the idea of an engineer inglobing local laws into descriptions which are specific to the device.

References

[De Kleer & Brown, 1984] Johan de Kleer, J.S. Brown. A qualitative physics based on confluences. Artificial Intelligence Vol.24 n°1-3, December 1984.
[De Kleer, 1984] Johan de Kleer. How Circuits Work. Artificial Intelligence Vol.24 n°1-3, December 1984.
[Raiman 1986] Olivier Raiman. Order of Magnitude Reasoning. AAAI86.
[Dague et al., 1987] P. Dague, P. Devès, O. Raiman. Troubleshooting: when modeling is the trouble. AAAI87.
[Dormoy, 1987] Jean-Luc Dormoy. Résolution qualitative: complétude, interprétation physique et contrôle. Mise en oeuvre dans un langage à base de règles: BOOJUM. Paris 6 University Doctoral Thesis, December 1987.
[Dormoy, 1988] Jean-Luc Dormoy. Controlling Qualitative Resolution. AAAI88.
[Ritschard, 1983] Gilbert Ritschard. Computable qualitative comparative statics techniques. Econometrica Vol. 51 n°4, July 1983.
[Weld, 1986] Dan Weld. The use of aggregation in causal simulation. Artificial Intelligence Vol. 30.