ADAPTIVE PROFORMAS FOR ACCIDENT AND EMERGENCY MEDICINE

by

Bryan S. Todd

Oxford University Computing Laboratory
Programming Research Group

ADAPTIVE PROFORMAS FOR ACCIDENT AND EMERGENCY MEDICINE

by

Bryan S. Todd

Technical Report PRG-TR-4-97   Feb 1997

Oxford University Computing Laboratory
Programming Research Group
Wolfson Building
Parks Road
Oxford   OX1 3QD
England

Oxford University Computing Laboratory
Programming Research Group
Wolfson Building
Parks Road
Oxford OX1 3QD
England

Electronic mail: todd@comlab.ox.ac.uk (JANET)

Adaptive Proformas for Accident and Emergency Medicine

Bryan S. Todd

Summary

This paper presents a simple mathematical model of the computerized patient record. The model is focussed on the requirement that during entry of a case, the computer program should interactively generate a proforma that is adapted to the particular clinical problem. Thus only essential information is collected and displayed, the number of mouse clicks is minimized thus simplifying data-entry, and prompts and warnings are given that help avoid oversights and errors. These ideas are illustrated by the documentation of patients presenting to an accident and emergency department.

Contents

1 Proformas

Proformas are increasingly being used for the documentation of clinical details of patients, especially in accident and emergency departments. The advantage of a proforma is that it encourages collection of all relevant information and displays this information in a structured and uniform way. Better documentation reduces the risk of litigation, facilitates clinical audit (particularly if clinical data are entered directly into a computer), and leads to improved clinical decision-making. For example, use of structured data-collection forms has been shown to improve diagnostic accuracy in the management of acute abdominal pain [Ada86].

However, it is often difficult to design a truly universal proforma even for a quite limited condition. For example, the appropriate symptoms and signs to elicit in a case of ankle injury are largely determined by the apparent severity of the trauma. In the absence of gross deformity, it is important to know whether the patient can bear weight through the injured limb, and it is important to map out the precise areas of tenderness: this helps to determine whether an X-ray is indicated and, if so, which views to request. However, if there is an obvious fracture-dislocation, immediate reduction and splintage should be performed, any possible neurovascular damage should be carefully assessed, and then an X-ray should be taken of the full length of the tibia and fibula. The possible documentation even of a case of isolated ankle injury therefore varies widely. Furthermore, multiple injuries are common, and the management of one injury is often influenced by the management of concurrent injures. For example, if a patient has a minor head injury with transient loss of consciousness, a skull X-ray is not indicated, nor is it appropriate to give head-injury instructions, if the patient is to be admitted to hospital for other reasons (e.g. internal fixation of an ankle fracture).

Clearly if the potential benefits of using a proforma are to be fully realised for all patients presenting to accident and emergency departments then it is necessary to adapt the proforma to the particular clinical problem of each patient. This could be most readily achieved by generating the proforma interactively by computer as clinical data are entered, the final record being printed out ready for signature: see, for example, [Abe89, Mar92]. However, proforma generation would require a substantial knowledge-base that would guide sensible formulation of the clinical problem and acquisition of relevant information. Several alternative representations for the knowledge base seem possible: flowcharts, production rules, object-oriented hierarchies. Nevertheless, the design of a suitable computer program is non-trivial. In the rest of this paper we begin this task by presenting in the 'Z' specification language [Spi88] a simple and abstract mathematical model of interactive proforma generation.

2 Formalism

A proforma is a vehicle for documenting a case. The vocabulary of medical terms (symptoms, signs, investigations, and so forth) is huge, but fortunately only a small number of these variables are relevant to any particular patient. Documenting the case entails instantiating those variables. This is easily formalized. Let us introduce the following two primitive sets.

$$\begin{aligned} \mathrm{Var} &= \text{`the set of all medical variables'} \\ \mathrm{Val} &= \text{`the set of all possible values for the variables'} \end{aligned}$$

We regard a *case* record as a partial mapping from variables to values. Let 'Case' be the set of all case records, whether feasible or infeasible.

$$\mathrm{Case} \;\widehat{=}\; \mathrm{Var} \nrightarrow \mathrm{Val} \tag{1}$$

For example, the following are three cases (grossly simplified for illustrative purposes) that are commonly encountered in accident and emergency departments.

$$\begin{aligned} c_0 \;\widehat{=}\; & \{\mathrm{injury} \mapsto \mathrm{wrist}, \\ & \ \mathrm{snuffbox} \mapsto \mathrm{non\text{-}tender}, \\ & \ \mathrm{x\text{-}ray\_of\_wrist} \mapsto \mathrm{normal}, \\ & \ \mathrm{diagnosis} \mapsto \mathrm{sprained\_wrist}, \\ & \ \mathrm{treatment} \mapsto \mathrm{tubigrip}\} \\[1em] c_1 \;\widehat{=}\; & \{\mathrm{injury} \mapsto \mathrm{wrist}, \\ & \ \mathrm{snuffbox} \mapsto \mathrm{tender}, \\ & \ \mathrm{x\text{-}ray\_of\_wrist} \mapsto \mathrm{normal}, \\ & \ \mathrm{diagnosis} \mapsto \mathrm{suspected\_scaphoid\_fracture}, \\ & \ \mathrm{treatment} \mapsto \mathrm{scaphoid\_plaster}\} \\[1em] c_2 \;\widehat{=}\; & \{\mathrm{injury} \mapsto \mathrm{ankle}, \\ & \ \mathrm{lateral\_malleolus} \mapsto \mathrm{tender}, \\ & \ \mathrm{x\text{-}ray\_of\_ankle} \mapsto \mathrm{normal}, \\ & \ \mathrm{diagnosis} \mapsto \mathrm{sprained\_ankle}, \\ & \ \mathrm{treatment} \mapsto \mathrm{elevation\_and\_tubigrip}\} \end{aligned}$$

A conventional proforma for a particular condition (e.g. acute abdominal pain) provides a fixed list of variables, all of which must be recorded. An adaptive proforma, one suitable for *any* condition, modifies the list of variables according to their recorded values. Nevertheless, a fundamental and obvious property of an adaptive proforma is that, at any point during data entry, the further information sought should be completely determined by the information gathered thus far. We refer to this property as 'determinism'.

Formally, we identify a *proforma* with the subset of cases that it can document. A comprehensive proforma is one that includes all medically possible cases that might present and an ideal one excludes all impossible cases. Considerable professional knowledge, judgement, and experience is needed to construct a proforma that is sufficiently close to the ideal to be useful, but we ignore these issues.

At the present level of abstraction determinism means that, for any set of two or more cases that can be documented by a given proforma, the set is 'partitionable' on the variables that are recorded in all the cases. That is to say, at least one of those variables differs in value between two or more of the cases. If this were not true then during data entry, if only those variables were recorded, it would be impossible to say which variable or variables should be recorded next. Let the symbol 'Proforma' stand for the set of all valid proformas.

$$\text{Proforma} \;\widehat{=}\; \{F : \mathbb{P}\,\text{Case} \mid \forall G : \mathbb{P}\,F \bullet \bigcap \mathsf{dom}\,(\!| G |\!) = \mathsf{dom}\,(\bigcap G) \Rightarrow \#G \leq 1\} \qquad (2)$$

Notice that determinism is captured by the systematic failure of dom to distribute through intersection. For each set of cases G, the term '$\bigcap \mathsf{dom}\,(\!| G |\!)$' is the set of variables that are recorded in every case in G, while the term '$\mathsf{dom}\,(\bigcap G)$' is the subset of these that have identical values in every case in G. Equality of the two terms means that G is not partitionable. This single definition captures all the required properties of a proforma, and the rest of this paper examines these in detail.

Notice that any set of total functions satifies the stated condition for a proforma, as does any set of partial functions which all share exactly the same domain. These correspond to conventional proformas, such as those printed on forms, whose set of variables is fixed. However, in this paper we are interested primarily in adaptive (or dynamic) proformas in which the relevant variables are chosen according to the nature of the case. We illustrate such a proforma with a trivial example. Suppose that the three cases c_0, c_1, and c_2 above were the only cases that could ever be recorded. It is possible to document precisely these cases by an adaptive proforma F_{EX} defined by

$$F_{\text{EX}} \;\widehat{=}\; \{c_0, c_1, c_2\}$$

In order to show that F_{EX} is a valid proforma we check each subset for the non-distributive property described in Equation 2. Consider F_{EX} itself.

$$\begin{aligned}
\bigcap \mathsf{dom}\,(\!| \{c_0, c_1, c_2\} |\!) &= \bigcap \{\mathsf{dom}\, c_0,\ \mathsf{dom}\, c_1,\ \mathsf{dom}\, c_2\} \\
&= \mathsf{dom}\, c_0 \,\cap\, \mathsf{dom}\, c_1 \,\cap\, \mathsf{dom}\, c_2 \\
&= \{\text{injury, snuffbox, x-ray_of_wrist, diagnosis, treatment}\} \\
&\quad \cap \{\text{injury, snuffbox, x-ray_of_wrist, diagnosis, treatment}\} \\
&\quad \cap \{\text{injury, lateral_malleolus, x-ray_of_ankle, diagnosis, treatment}\} \\
&= \{\text{injury, diagnosis, treatment}\} \\
&\neq \{\} \\
&= \mathsf{dom}\,\{\} \\
&= \mathsf{dom}\,(c_0 \cap c_1 \cap c_2) \\
&= \mathsf{dom}\,(\bigcap \{c_0, c_1, c_2\})
\end{aligned}$$

Similarly, inequality can be demonstrated for all other subsets of F_{EX} larger than singletons. This means that F_{EX} can be implemented as a deterministic procedure that prompts the user for the values of relevant variables until one of the three cases has been entered. We show this in the next section. Before doing so though, we examine some of the properties of proformas.

2.1 Some Properties

Firstly, notice that the empty set is a valid proforma.

Lemma 1

$$\vdash \{\} \in \mathrm{Proforma}$$

Proof

$$\forall G : \mathrm{P}\{\} \bullet \; \#G \leq 1$$

□

However, the empty set is quite useless as a proforma since, by definition, it is unable to document any case at all! We would never wish to implement the empty proforma.

Secondly, notice that if a case can be documented by a given proforma then no strict superset can also be documented by the same proforma. Termination of data-entry is therefore also deterministic. This is one reason why determinism is so important.

Lemma 2

$$F : \mathrm{Proforma}; c, d : \mathrm{Case} \mid \{c, d\} \subseteq F \vdash c \not\subset d$$

Proof
The proof is given on Page 12 of the Appendix. □

Thirdly, notice that the set of all valid proformas is subset-closed.

Lemma 3

$$E, F : \mathrm{P}\,\mathrm{Case} \mid E \subseteq F \wedge F \in \mathrm{Proforma} \vdash E \in \mathrm{Proforma}$$

Proof

$$G \in \mathrm{P}\,E \Rightarrow G \in \mathrm{P}\,F$$

□

In particular, this means that a proforma F can be strengthened by intersection with any set of cases. This property should make implementation easier by decomposing the task. For example, first a weak proforma is implemented that inputs variables in a sensible order, but which allows medically-infeasible combinations of values. Then the proforma is placed in an environment that restricts cases to valid combinations of values [Hoa85].

On a technical note, in the definition of a proforma (Equation 2) set G needs to range over all subsets of F of any size and cannot be restricted to those with two members. The following is a counter-example.

$$F_{\mathrm{CX}} \;\widehat{=}\; \{\{x \mapsto 0, y \mapsto 0\}, \{y \mapsto 1, z \mapsto 0\}, \{x \mapsto 1, z \mapsto 1\}\}$$

Every subset of F_{CX} with two elements is partitionable but no variable is common to all three cases. This means that, starting from the empty case record, no variable is necessarily relevant. Notice that with the stronger condition (Equation 2) $F_{\mathrm{CX}} \notin \mathrm{Proforma}$ because F_{CX} itself is not partionable: $\bigcap \mathsf{dom} (\! | F_{\mathrm{CX}} | \!) = \mathsf{dom} (\bigcap F_{\mathrm{CX}}) = \{\}$.

3 Data-Entry

Consider now the process of data-entry. At any point during this process some case has been partially entered. A proforma F accepts entry of a case c only if c has some possible valid completion. Let '$F\ \underline{accepts}\ c$' denote this relationship.

$$\_\ \underline{accepts}\ \_ : \mathrm{Proforma} \leftrightarrow \mathrm{Case} \tag{3}$$

$$\begin{array}{l} \forall F : \mathrm{Proforma};\, c : \mathrm{Case} \bullet \\ \qquad F\ \underline{accepts}\ c \;\Leftrightarrow\; \exists d : F \bullet c \subseteq d \end{array}$$

Notice that any proforma (except the empty one) accepts the empty case. This is just as well since the empty case is always the initial state of a case record during data-entry.

Lemma 4

$$F : \mathrm{Proforma} \;\mid\; F \neq \{\} \;\vdash\; F\ \underline{accepts}\ \{\}$$

Proof

$$\exists d : F \bullet \{\} \subseteq d$$

□

Furthermore, any proforma accepts all cases that it is able to document. This too is just as well since these are the possible final states of a case record during data-entry.

Lemma 5

$$F : \mathrm{Proforma}; c : \mathrm{Case} \;\mid\; c \in F \;\vdash\; F\ \underline{accepts}\ c$$

Proof

$$\exists d : F \bullet c \subseteq d$$

□

Indeed, more generally, the set of cases that a proforma accepts is subset-closed. This means that at any point during data-entry, one or more facts can be retracted.

Lemma 6

$$F : \mathrm{Proforma}; c, d : \mathrm{Case} \mid c \subseteq d \land F \ \underline{\mathit{accepts}}\ d \vdash F \ \underline{\mathit{accepts}}\ c$$

Proof

$$\exists d : F \bullet c \subseteq d$$

$\square$

Therefore, if any item of entered data is changed, the whole entry procedure is automatically restructured, but none of the other data needs re-entry.

3.1 Next Variables

Suppose that during data-entry into a proforma F, the current state of the case record is c. Let 'nextvars $F\,c$' be the set of uninstantiated variables that are provably relevant to the case. These are the variables that are relevant to all possible completions of c.

$$\begin{aligned} &\mathrm{nextvars} : \mathrm{Proforma} \to \mathrm{Case} \to \mathbb{P}\,\mathrm{Var} \\ &\forall F : \mathrm{Proforma}; c : \mathrm{Case} \bullet \\ &\quad \mathrm{nextvars}\, F\, c \ = \ \bigcap \{ d : F \mid c \subseteq d \bullet \mathsf{dom}\,(d - c) \} \end{aligned} \tag{4}$$

Notice that the next relevant variables never include any that have already been instantiated.

Lemma 7

$$F : \mathrm{Proforma}; c : \mathrm{Case} \mid F \ \underline{\mathit{accepts}}\ c \vdash (\mathrm{nextvars}\, F\, c) \cap (\mathsf{dom}\, c) = \{\}$$

Proof
The proof is given on Page 12 of the Appendix. $\square$

Furthermore, if case c has some possible completion d, then the next relevant variables include only those of d. Instantiation of possibly irrelevant variables is thus avoided. Furthermore, this also means that it is immaterial in which order the variables in 'nextvars $F\,c$' are instantiated.

Lemma 8

$$F : \mathrm{Proforma}; c, d : \mathrm{Case} \mid c \subseteq d \land d \in F \vdash \mathrm{nextvars}\, F\, c \subseteq \mathsf{dom}\, d$$

Proof

$$\bigcap \{ d : F \mid c \subseteq d \bullet \mathsf{dom}\,(d - c) \} \ \subseteq \ \mathsf{dom}\, d$$

$\square$

Lastly, when (and only when) data-entry is complete no further variables are relevant.

Lemma 9

$$F : \text{Proforma}; c : \text{Case} \mid F\ \underline{\mathit{accepts}}\ c \vdash \text{nextvars}\, F\, c = \{\} \Leftrightarrow c \in F$$

Proof
The proof is given on Page 13 of the Appendix. □

3.2 Possible Values

Not only does a proforma determine which uninstantiated variables are relevant given any state of the case record, but it also determines which values are possible for those variables. Given proforma F, partial case record c, and variable v, let 'possvals $F\ c\ v$' be the set of possible values for v. These are the values v takes in every possible completion of c.

$$\text{possvals} : \text{Proforma} \to \text{Case} \to \text{Var} \to \mathbb{P}\,\text{Val} \tag{5}$$

$$\begin{aligned}&\forall F : \text{Proforma}; c : \text{Case}; v : \text{Var} \bullet \\ &\quad \text{possvals}\, F\ c\ v = \{u : \text{Val} \mid \exists d : F \bullet c \subseteq d \wedge (v \mapsto u) \in d\}\end{aligned}$$

Notice that during data-entry, each *instantiated* variable has only one possible value, the one that has been assigned to the variable.

Lemma 10

$$F : \text{Proforma}; c : \text{Case}; v : \text{Var} \mid F\ \underline{\mathit{accepts}}\ c \wedge v \in \mathsf{dom}\, c \vdash \text{possvals}\, F\ c\ v = \{c(v)\}$$

Proof
The proof is given on Page 14 of the Appendix. □

3.3 Some Examples

Consider use of the proforma F_{EX} (defined on Page 3) to document a case. The initial state of the case record is empty. By application of the function 'nextvars' we discover which variables are initially relevant.

$$\begin{aligned}\text{nextvars}\, F_{\text{EX}}\, \{\} &= \bigcap \{d : \{c_0, c_1, c_2\} \mid \{\} \subseteq d \bullet \mathsf{dom}\,(d - \{\})\} \\ &= \bigcap \{d : \{c_0, c_1, c_2\} \bullet \mathsf{dom}\, d\} \\ &= \bigcap \{\mathsf{dom}\, c_0, \mathsf{dom}\, c_1, \mathsf{dom}\, c_2\} \\ &= \mathsf{dom}\, c_0 \cap \mathsf{dom}\, c_1 \cap \mathsf{dom}\, c_2 \\ &= \{\text{injury, snuffbox, x-ray\_of\_wrist, diagnosis, treatment}\} \\ &\quad \cap \{\text{injury, snuffbox, x-ray\_of\_wrist, diagnosis, treatment}\} \\ &\quad \cap \{\text{injury, lateral\_malleolus, x-ray\_of\_ankle, diagnosis, treatment}\} \\ &= \{\text{injury, diagnosis, treatment}\}\end{aligned}$$

The user is therefore prompted to enter the site of injury, the diagnosis, and the treatment, in any order. By application of the function 'possvals' we discover the possible values for each of these variables. This should make it easier for the user to choose.

$$\begin{aligned}
\text{possvals}\ F_{\text{EX}}\ \{\}\ \text{injury} &= \{u : \text{Val} \mid \exists d : \{c_0, c_1, c_2\} \bullet \{\} \subseteq d \wedge (\text{injury}, u) \in d\} \\
&= \{u : \text{Val} \mid \exists d : \{c_0, c_1, c_2\} \bullet (\text{injury}, u) \in d\} \\
&= \{\text{wrist, ankle}\}
\end{aligned}$$

Similarly,

$$\begin{aligned}
\text{possvals}\ F_{\text{EX}}\ \{\}\ \text{diagnosis} &= \{\text{sprained_wrist}, \text{suspected_scaphoid_fracture}, \text{sprained_ankle}\} \\
\text{possvals}\ F_{\text{EX}}\ \{\}\ \text{treatment} &= \{\text{tubigrip, scaphoid_plaster, tubigrip_and_elevation}\}
\end{aligned}$$

Suppose that the user enters 'wrist' as the site of the injury. Immediately more variables become relevant, even before the diagnosis and treatment are recorded.

$$\text{nextvars}\ F_{\text{EX}}\ \{\text{injury} \mapsto \text{ankle}\} = \{\text{snuffbox, x-ray\_of\_wrist, diagnosis, treatment}\}$$

Thus the user is also prompted for the values of two further variables: 'snuffbox' and 'x-ray_of_wrist'. The possible values for all four relevant variables are

$$\begin{aligned}
\text{possvals}\ F_{\text{EX}}\{\text{injury} \mapsto \text{ankle}\}\ \text{snuffbox} &= \{\text{tender, non-tender}\} \\
\text{possvals}\ F_{\text{EX}}\{\text{injury} \mapsto \text{ankle}\}\ \text{x-ray_of_wrist} &= \{\text{normal}\} \\
\text{possvals}\ F_{\text{EX}}\{\text{injury} \mapsto \text{ankle}\}\ \text{diagnosis} &= \{\text{sprained_wrist}, \text{suspected_scaphoid_fracture}\} \\
\text{possvals}\ F_{\text{EX}}\{\text{injury} \mapsto \text{ankle}\}\ \text{treatment} &= \{\text{tubigrip, scaphoid_plaster}\}
\end{aligned}$$

Notice that the sets of possible diagnoses and treatments have diminished in the light of the new information about the case. Also an assumption is made that the X-ray is normal. In practice this assumption is usually correct, but obviously any proforma intended for actual use would have to handle all the common abnormalities that are seen on X-ray.

Suppose that the user enters the information that there is tenderness in the anatomical snuffbox. Only 'diagnosis' and 'treatment' now remain relevant, and their values are determined by F_{EX}.

$$\begin{aligned}
\text{nextvars}\ F_{\text{EX}}\ c &= \{\text{diagnosis, treatment}\} \\
\text{possvals}\ F_{\text{EX}}\ c\ \text{diagnosis} &= \{\text{suspected_scaphoid_fracture}\} \\
\text{possvals}\ F_{\text{EX}}\ c\ \text{treatment} &= \{\text{scaphoid_plaster}\} \\
\underline{\textit{where}}\ c &\mathrel{\widehat{=}} \{\text{injury} \mapsto \text{ankle, snuffbox} \mapsto \text{tender, x-ray_of_wrist} \mapsto \text{normal}\}
\end{aligned}$$

Notice that inference about a variable's possible values helps both to avoid unnecessary mouse clicks (in the case that there is only one possible value) and to avoid inappropriate diagnoses (e.g. 'sprained wrist') and treatments (e.g. a tubigrip bandage instead of a scaphoid plaster).

4 A Demonstrator Prototype

In order to investigate the feasibility of the approach delineated in this paper, we implemented a simple demonstrator program. Only ankle injuries were dealt with in any detail; the documentation of any other condition terminates after the recording of just a few basic facts (e.g. name, age, GP, category, diagnosis, disposal). A total of 41 variables and 212 values were incorporated into the program (i.e. #Var = 41 and #Val = 212). All variables were considered categorical: any full implementation would of course need to deal with numerical variables and variables whose values are arbitrary strings *etc.*

The knowledge base comprises two parts. The first part defines a 'comprehensive' proforma F_{CP} by means of a relation $\underline{rel}$ between facts (i.e. variable-value pairs) and variables such that $(v \mapsto u)\ \underline{rel}\ w$ means that if variable v is recorded as having value u then variable w becomes relevant. This relation must be acyclic in the sense that the variables can be totally ordered such that whenever $(v \mapsto u)\ \underline{rel}\ w$ variable w is strictly above variable v. If the ordering is made explicit in the knowledge base (in our experience this makes it easier to specify the relation) then it is computationally efficient and trivial to check for acyclicity. It can easily be shown that any such acyclic relation defines a valid proforma.

The second part of the knowledge base constrains F_{CP} to a smaller proforma F_{ID} that approaches the 'ideal' by eliminating impossible combinations of facts. For simplicity only impossible pairs rather than larger inconsistent sets were considered. This part of the knowledge base thus comprises a symmetric exclusion relation $\underline{exc}$ such that $(v \mapsto u)\ \underline{exc}\ (w \mapsto t)$ means that facts $(v \mapsto u)$ and $(w \mapsto t)$ are mutually exclusive. Thus

$$F_{\mathrm{ID}} \quad \widehat{=} \quad \{c : F_{\mathrm{CP}} \mid (\underline{exc}) \cap (c \times c) = \{\}\}$$

Lemma 3 (Page 4) guarantees that F_{ID} is a valid proforma whatever the nature of the relation $\underline{exc}$. For clarity, exclusion mappings concerning the same pair of variables were grouped together in the knowledge base: a total of 41 such groups were included.

Given any partial case c, computation of 'nextvars $F_{\mathrm{ID}}\ c$' entails identifying all new variables made directly relevant ($\underline{rel}(\!|c|\!) - \mathsf{dom}\, c$) and then repeatedly including each additional variable v such that v is relevant in every valid instantiation of the other relevant variables[1]. This involves repeated tautology checking of the logical theory embodied by the exclusion relation. For this reason only pairs of mutually exclusive facts were considered: tautology checking of such a logical theory is computationally equivalent to the 2-SAT problem and so can be performed easily and efficiently by an adaptation of the transitive closure algorithm.

Three windows are displayed on the screen. In the first is an initially-empty list of facts recorded about the patient: these are sorted and displayed in a medically-familiar order (symptoms, signs, investigations, diagnosis, treatment, disposal). In the second window is

[1]For expediency, after an item is entered or the cursor is moved, the demonstrator program makes only a single pass rather than iterating until all relevant variables are found and so sometimes does not find every relevant variable immediately.

the set of relevant but as yet uninstantiated set of variables. In the third window is the set of possible values for whichever variable in the second window is currently highlighted by the cursor (which can be moved up and down at will). Selection of a value from the third window instantiates the highlighted variable in the second window and transfers this fact to the first window. As a result, some newly relevant variables may be added to the second window. Use of the program entails repeatedly instantiating variables in the second window until none remain: the first window by then contains the complete documentation of the case. Members of Consultant A&E staff who have seen the demonstrator program have expressed their strong support for the suitability of the paradigm.

For future versions of the program we will explore alternative knowledge representations. Although we experienced no great difficulty in specifying a relevancy relation, we did feel that in a few instances it would have been helpful to have had a more flexible representation. For example, in the context of ankle injury, 'social history' is relevant really only if the patient is being discharged home *and* is elderly: this can be expressed only if the relation is extended to allow conjunctions of facts in its domain. Similarly, the notion of an exclusion relation proved a little too restrictive. For example, if a patient is sent home with a fracture of the ankle then the patient needs a referral to the New Patient Clinic for the following morning. This cannot be expressed by a purely binary exclusion relation on facts. In future versions of the program we intend to represent the logical theory by a large conjunction of arbitrarily complicated logical expressions, each involving only a relatively small number of variables (say ten). Modern tautology checkers (e.g. [Sta89]) may be able to handle such a theory efficiently. If not, one simple solution [Tod97] is to adapt the Lauritzen-Spiegelhalter algorithm for propagating evidence through bayesian networks [Lau88].

5 Summary

In this paper we have presented a simple model of an adaptive proforma for the documentation of case records. The model is based on a single definition (Equation 2) that captures the requirement that variables relevant to a case are determined by the values of variables already recorded. This is expressed in terms of a non-distributive property of the domain operator through set intersection. The remainder of the paper explores the consequences of this definition and shows the model to have intuitive and desirable properties. The advantage of a simple abstract model is that it allows for a wide range of possible implementations (flowcharts, rule-based systems, object hierarchies) each of which can be verified against the original definition and thus shown to have the intended properties. A demonstrator prototype using a rudimentary knowledge representation has shown the feasibility of the approach and the suitability of the paradigm for the intended application.

Acknowledgement

This work has been partly supported by an EPSRC Advanced Fellowship in Information Technology (B/ITF/269). I am grateful to Tony Hoare for his comments on earlier drafts of this paper.

References

[Abe89] **Abendroth TW, Greenes RA.** A clinical algorithm processor: enabling flow charts to organize a variety of physician tasks. *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care, Washington DC, USA* (1989) 983–4.

[Ada86] **Adams ID, Chan M, Clifford PC, Cooke WM, Dallos V, de Dombal FT, Edwards MH, Hancock DM, Hewett DJ, McIntyre N, Somerville PG, Spiegelhalter DJ, Wellwood J, Wilson DH.** Computer-aided diagnosis of acute abdominal pain: a multicentre study. *British Medical Journal* (1986) **293** 800–4.

[Hoa85] **Hoare CAR.** *Communicating Sequential Processes* (1985) Prentice-Hall International UK Ltd: London.

[Lau88] **Lauritzen SL, Spiegelhalter DJ.** Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B* (1988) **50** 157–224.

[Mar92] **Margolis CZ, Warshawsky SS, Goldman L, Dagan O, Wirtschafter D, Pliskin JS.** Computerized algorithms and pediatricians' management of common problems in a community clinic. *Academic Medicine* (1992) **67** *iv* 282–4.

[Spi88] **Spivey JM.** *The Z Notation: A Reference Manual* (1988) Prentice-Hall.

[Sta89] **Stålmarck G.** A system for determining propositional logic theorems by applying values and rules to triplets that are generated from a formula. *U.S. patent 5 276 897* (1989, approved 1994).

[Tod97] **Todd BS.** Efficient evaluation of boolean expressions. *Technical Report PRG-TR-3-97, Oxford University Computing Laboratory* (1997).

Appendix: Proofs of Lemmas

Lemma 2 (Page 4)

$$F : \text{Proforma}; c, d : \text{Case} \;\mid\; \{c, d\} \subseteq F \;\vdash\; c \not\subset d$$

Proof
We prove the result through contradiction. Suppose that $c \subset d$. Since therefore $c \neq d$ we have that $\#\{c, d\} = 2$. However, from the definition of Proforma (Equation 2),

$$\forall G : \mathbb{P}\, F \;\mid\; \#G \geq 2 \bullet\; \bigcap \mathsf{dom}\,(\!| G |\!) \neq \mathsf{dom}\,(\bigcap G)$$

Therefore, putting $G = \{c, d\}$,

$$\bigcap \mathsf{dom}\,(\!| \{c, d\} |\!) \neq \mathsf{dom}\,(\bigcap \{c, d\})$$

However, we now obtain equality of the two terms also, thus establishing a contradiction.

$$\begin{array}{rcll}
\bigcap \mathsf{dom}\,(\!| \{c, d\} |\!) & = & \bigcap \{\mathsf{dom}\, c,\ \mathsf{dom}\, d\} & \text{Property of relational image.} \\
& = & \mathsf{dom}\, c \cap \mathsf{dom}\, d & \text{Property of generalized intersection.} \\
& = & \mathsf{dom}\, c & \text{Since } c \subset d. \\
& = & \mathsf{dom}\,(c \cap d) & \text{Since } c \subset d. \\
& = & \mathsf{dom}\,(\bigcap \{c, d\}) & \text{Property of generalized intersection.}
\end{array}$$

Therefore the original supposition ($c \subset d$) must be false. □

Lemma 7 (Page 6)

$$F : \text{Proforma}; c : \text{Case} \;\mid\; F\ \underline{accepts}\ c \;\vdash\; (\text{nextvars}\, F\, c) \cap (\mathsf{dom}\, c) = \{\}$$

Proof

$$\begin{array}{rcll}
(\text{nextvars}\, F\, c) \cap (\mathsf{dom}\, c) & = & \bigcap \{d : F \mid c \subseteq d \bullet \mathsf{dom}\,(d - c)\} \cap \mathsf{dom}\, c & \text{Defn nextvars.} \\
& = & \bigcap \{d : F \mid c \subseteq d \bullet \mathsf{dom}\,(d - c) \cap \mathsf{dom}\, c\} & \text{Set theory.} \\
& = & \bigcap \{d : F \mid c \subseteq d \bullet \{\}\} & \text{Set theory.} \\
& = & \bigcap \{\{\}\} & F\ \underline{accepts}\ c. \\
& = & \{\} & \text{Set theory.}
\end{array}$$

□

Lemma 9 (Page 7)

$$F : \text{Proforma}; c : \text{Case} \mid F \ \underline{\textit{accepts}}\ c \vdash \text{nextvars}\, F\, c = \{\} \Leftrightarrow c \in F$$

Proof
Let us define

$$G \mathrel{\widehat{=}} \{d : F \mid c \subseteq d\}$$

Notice that $G \neq \{\}$ since F $\underline{\textit{accepts}}$ c. We now complete the proof by examining separately the two alternative cases $\#G = 1$ and $\#G \geq 2$.

<u>Case: $\#G = 1$</u>
Let d be the unique case such that $G = \{d\}$. It follows from the definition of nextvars that

$$\text{nextvars}\, F\, c = \mathsf{dom}\,(d - c)$$

Therefore

$$\begin{array}{rcll} \text{nextvars}\, F\, c = \{\} & \Leftrightarrow & \mathsf{dom}\,(d - c) = \{\} & \text{Equality above.} \\ & \Leftrightarrow & c = d & \text{Since } c \subseteq d \text{ (definition of } G\text{).} \end{array}$$

However,

$$\begin{array}{rcll} c = d & \Rightarrow & c \in F & \text{Since } d \in F \text{ (definition of } G\text{).} \\ & \Rightarrow & c \in G & \text{Since } c \subseteq c. \\ & \Rightarrow & c = d & \text{Since } G = \{d\}. \end{array}$$

Therefore, since $c = d \Leftrightarrow c \in F$,

$$\text{nextvars}\, F\, c = \{\} \Leftrightarrow c \in F$$

<u>Case: $\#G \geq 2$</u>
It follows from the definition of G that

$$\mathsf{dom}\, c \subseteq \mathsf{dom}\,(\bigcap G) \subseteq \bigcap \mathsf{dom}\,(\!| G |\!)$$

Furthermore, since $G \subseteq F$, it follows from the definition of Proforma that

$$\mathsf{dom}\,(\bigcap G) \neq \bigcap \mathsf{dom}\,(\!| G |\!)$$

Therefore,

$$\mathsf{dom}\, c \subset \bigcap \mathsf{dom}\,(\!| G |\!)$$

Thus,

$$\begin{array}{rcll} \text{nextvars}\, F\, c & = & \bigcap \{d : F \mid c \subseteq d \bullet \mathsf{dom}\,(d - c)\,\} & \text{Definition of nextvars.} \\ & = & \bigcap \{d : F \mid c \subseteq d \bullet \mathsf{dom}\, d - \mathsf{dom}\, c\,\} & \text{Since } c \subseteq d. \\ & = & \bigcap \{d : F \mid c \subseteq d \bullet \mathsf{dom}\, d\,\} - \mathsf{dom}\, c & \text{Set theory.} \\ & = & \bigcap \mathsf{dom}\,(\!| \{d : F \mid c \subseteq d\} |\!) - \mathsf{dom}\, c & \text{Set theory.} \\ & = & \bigcap \mathsf{dom}\,(\!| G |\!) - \mathsf{dom}\, c & \text{Definition of } G. \\ & \neq & \{\} & \text{Result above.} \end{array}$$

Furthermore, if c were a member of F then, since $\#G \geq 2$, there must be at least one member d of G such that $c \neq d$. However, from the definition of G, we have that $d \in F$ and $c \subseteq d$, which implies that $c \subset d$. This is not possible (Lemma 2). Therefore, we must have that $c \notin F$. Thus

$$\text{nextvars}\, F\, c = \{\} \;\Leftrightarrow\; c \in F$$

holds trivially by virtue of the fact that both operands of $\Leftrightarrow$ are false. $\square$

Lemma 10 (Page 7)

$$F : \text{Proforma}; c : \text{Case}; v : \text{Var} \;\mid\; F\ \underline{\textit{accepts}}\ c \wedge v \in \mathsf{dom}\, c \;\vdash\; \text{possvals}\, F\, c\ v = \{c(v)\}$$

Proof

$$\begin{array}{rcll}
\text{possvals}\, F\, c\ v & = & \{u : \text{Val} \mid \exists d : F \mid c \subseteq d \wedge (v,u) \in d\} & \text{Definition of possvals.} \\
& = & \{u : \text{Val} \mid \exists d : F \mid c \subseteq d \wedge (v,u) \in c\} & \text{Since } v \in \mathsf{dom}\, c. \\
& = & \{u : \text{Val} \mid (v,u) \in c\} & \text{Since } F\ \underline{\textit{accepts}}\ c. \\
& = & \{c(v)\} & \text{Since } v \in \mathsf{dom}\, c.
\end{array}$$

$\square$