Inductive Situation Calculus

Marc Denecker
Department of Computer Science, K.U.Leuven,
Celestijnenlaan 200A, B-3001 Heverlee,
Belgium
email: marcd@cs.kuleuven.ac.be

Eugenia Ternovska
School of Computing Science,
Simon Fraser University,
Burnaby, BC, V5A 1S6, Canada
email: ter@cs.sfu.ca

Abstract

Temporal reasoning has always been a major test case for knowledge representation formalisms. In this paper, we develop an inductive variant of the situation calculus using the Logic for Non-Monotone Inductive Definitions (NMID). This logic has been proposed recently and is an extension of classical logic. It allows for a uniform represention of various forms of definitions, including monotone inductive definitions and non-monotone forms of inductive definitions such as iterated induction and induction over well-founded posets. In the NMID-axiomatisation of the situation calculus, fluents and causality predicates are defined by simultaneous induction on the well-founded poset of situations. The inductive approach allows us to solve the ramification problem for the situation calculus in a uniform and modular way. Our solution is among the most general solutions for the ramification problem in the situation calculus. Using previously developed modularity techniques, we show that the basic variant of the inductive situation calculus *without* ramification rules is equivalent to Reiter-style situation calculus.

Introduction and Preliminaries

The recently developed Logic for Non-Monotone Inductive Definitions (NMID) is an extension of classical logic that allows for uniform representation of various forms of definitions, including monotone inductive definitions and non-monotone forms of inductive definitions such as iterated induction and induction over well-founded posets (Denecker & Ternovska 2004).

Here, we demonstrate an application of NMID-logic. The aim is two-fold. First, we illustrate the role of NMID-logic and non-monotone inductive definitions for knowledge representation by presenting a variant of the situation calculus which we call *inductive situation calculus*. We show that ramification rules can be naturally modeled through a non-monotone iterated inductive definition. Second, we illustrate the use of our recently developed modularity techniques for NMID-logic in order to translate a theory of the inductive situation calculus into a classical logic theory of Reiter's situation calculus (Reiter 2001).

There are several points of interest in this experiment. The first one is our observation that complex non-monotone inductive definitions not only occur in mathematics, but also in common sense reasoning. In particular, we believe that the original Reiter-style situation calculus contains hidden forms of definitions which we explicitate in the inductive situation calculus. The second point is the fact that different forms of inductive definitions, which have a uniform representation in NMID-logic, can be formalised in classical (first- or second-order) logic as well, but not in a uniform way: different sorts of definitions require different formalisation. As a consequence, our formalisation is simpler, more uniform and more modular than Reiter-style situation calculus. The third point is that NMID-logic is closely connected to logic programming. In particular, it formally extends logic programming (LP) and abductive logic programming (ALP) under the well-founded semantics. The strong connection with LP and ALP can be exploited to build implementations of NMID-logic and of inductive situation calculus theories. Moreover, NMID-logic has an important advantage as a knowledge representation language — contrary to logic programming, it does not automatically impose the domain closure assumption (although the domain closure axiom can be expressed if needed).

In the remaining part of this section we briefly describe NMID-logic, the modularity theorem, and some techniques for translating NMID-logic theories into classical logic. We also review the more traditional variant of the situation calculus similar to (Reiter 2001). In the rest of the paper, we present the formalism of the inductive situation calculus, address the ramification problem and consider a detailed example.

NMID-Logic

First, we present an extension of classical logic with non-monotone inductive definitions. This work extends previous work of (Denecker 2000). A more detailed exposition can be found in (Denecker & Ternovska 2004). A new binary connective $\leftarrow$ is called the *definitional implication*. A *definition* Δ is a set of rules of the form $\forall \bar{x}\ (X(\bar{t}) \leftarrow \varphi)$, where $\bar{x}$ is a tuple of object variables, X is a predicate symbol (i.e., a predicate constant or variable) of some arity r, $\bar{t}$ is a tuple of terms of length r of the vocabulary τ, φ is an arbitrary first-order formula of τ. The definitional implication $\leftarrow$ must be distinguished from material implication.

Note that in front of rules, we allow only universal quantifiers. In the rule $\forall \bar{x}\ (X(\bar{t}) \leftarrow \varphi)$, $X(\bar{t})$ is called the *head*

and φ is the *body* of the rule. A *defined symbol* of Δ is a relation symbol that occurs in the head of at least one rule of Δ; other relation, object and function symbols are called *open*. We use notation $\phi(x_1, \ldots x_n)$ to emphasize that symbols $x_1, \ldots x_n$ are distinct and are free in ϕ. Let τ be a vocabulary including all free symbols of Δ. The subset of defined symbols of definition Δ is denoted τ^{d}_Δ. The set of open symbols of Δ in τ is denoted τ^{o}_Δ. The sets τ^{d}_Δ and τ^{o}_Δ form a partition of τ, i.e., $\tau^{\mathrm{d}}_\Delta \cup \tau^{\mathrm{o}}_\Delta = \tau$, and $\tau^{\mathrm{d}}_\Delta \cap \tau^{\mathrm{o}}_\Delta = \emptyset$.

A *well-formed formula* of the Logic for Non-Monotone Inductive Definitions, briefly a NMID-formula, is defined by the following (monotone) induction:

1. If X is an n-ary predicate symbol, and $t_1, \ldots, t_n$ are terms then $X(t_1, \ldots, t_n)$ is a formula.
2. If Δ is a definition then Δ is a formula.
3. If ϕ, ψ are formulas, then so is $(\phi \wedge \psi)$.
4. If ϕ is a formula, then so is $(\neg\phi)$.
5. If ϕ is a formula, then $\exists\sigma\ \phi$ is a formula (σ can be either a first- or second-order symbol).

A formula ϕ is an *NMID-formula over vocabulary* τ if the set of free symbols of ϕ is a subset of τ.

As an example, in the language of the natural numbes, the following formula expresses that there is a set which is the least set containing 0 and closed under taking successor numbers, and which contains all domain elements. It is equivalent to the standard induction axiom and with the domain closure axiom:

$$\exists N \left[\left\{ \begin{array}{l} \forall x(N(x) \leftarrow x = 0), \\ \forall x(N(s(x)) \leftarrow N(x)) \end{array} \right\} \wedge \forall x\ N(x) \right].$$

Note that this formula contains an existential quantification over the second-order variable N. This can be avoided by skolemising N and using a new predicate constant instead. In fact, all examples of second-order quantification that appear in this paper, are of the same kind as in this example and can be eliminated in the same way, by skolemisation of the existentially quantified second-order variable.

The semantics of the NMID-logic is an extension of classical logic semantics with the well-founded semantics from logic programming (Van Gelder 1993; Fitting 2003; Denecker, Bruynooghe, & Marek 2001). We now briefly describe this semantics. We assume familiarity of the reader with the semantics of classical logic. For more detail, we refer to (Denecker & Ternovska 2004). A structure I of a vocabulary τ consists of a domain $dom(I)$, and for each symbol $\sigma \in \tau$ a *value* σ^I in $dom(I)$, i.e., a domain element for an object symbol, a function for a function symbol and a relation for a predicate symbol of the corresponding arity. The value t^I of a term t in I is defined by the standard recursion.

We first define the well-founded model of a definition Δ extending a τ^{o}_Δ-structure I_{o}. For each defined symbol X of Δ, we define

$$\varphi_X(\bar{x}) := \exists\bar{y}_1\ (\bar{x} = \bar{t}_1 \wedge \varphi_1) \vee \cdots \vee \exists\bar{y}_m\ (\bar{x} = \bar{t}_m \wedge \varphi_m),$$

where $\bar{x}$ is a tuple of new variables, and $\forall\bar{y}_1\ (X(\bar{t}_1) \leftarrow \varphi_1)$, $\ldots$, $\forall\bar{y}_m\ (X(\bar{t}_m) \leftarrow \varphi_m)$ are the rules of Δ with X in the head. For every defined symbol Y, we introduce a new relation symbol Y' of the same arity. We obtain φ'_X from $\varphi_X(\bar{x})$ by substituting Y' for each negative occurrence of each defined symbol Y.

For any pair of τ-structures I, J extending I_{o}, define I_J as the extension of I_{o} which interprets each defined symbol X of Δ as X^I, the value of X in I, and each new symbol X' as X^J, the value of X in J. The basis of the construction of the well-founded model extending I_{o} is the operator T_Δ which maps pairs I, J of extensions of I_{o} to a structure I', also extending I_{o}, such that for each defined symbol X, $X^{I'} := \{\bar{a} \mid I_J \models \varphi'_X[\bar{a}]\}$. Thus, the operator T_Δ evaluates positive occurrences of defined symbols in rule bodies by I, and negative occurrences of defined symbols by J.

In the lattice of τ-structures extending I_{o}, the operator T_Δ is monotone in its first argument and anti-monotone in its second argument. Define the *stable*[1] *operator* ST_Δ as follows: $ST_\Delta(J) := \mathit{lfp}(\mathsf{T}_\Delta(\cdot, J))$. This stable operator is anti-monotone, hence its square is monotone and has a least and largest fixpoint. We define ${I_{\mathrm{o}}}^{\Delta\downarrow} := \mathit{lfp}(ST^2_\Delta)$, and ${I_{\mathrm{o}}}^{\Delta\uparrow} := \mathit{gfp}(ST^2_\Delta)$.

For an intuitive explanation of the well-founded semantics and an argument why it formalises different forms of inductive definitions, we refer to (Denecker, Bruynooghe, & Marek 2001).

Definition 1. *Definition Δ is* total *in τ^{o}_Δ-structure I_{o} if ${I_{\mathrm{o}}}^{\Delta\downarrow} = {I_{\mathrm{o}}}^{\Delta\uparrow}$. When this is the case, ${I_{\mathrm{o}}}^{\Delta\downarrow}$ (or ${I_{\mathrm{o}}}^{\Delta\uparrow}$) is called the* Δ-extension of I_{o} *and is abbreviated as ${I_{\mathrm{o}}}^{\Delta}$. More generally, Δ is* total in a structure K_{o} *interpreting a subset of τ^{o}_Δ if Δ is total in each τ^{o}_Δ-structure extending K_{o}.*

The aim of an inductive definition is to *define* its defined symbols. Therefore, a natural quality requirement for a definition is that it is total.

Below, $I[\sigma : v]$ denotes the structure obtained from I by assigning the value v to the symbol σ.

Definition 2 (ϕ true in structure I). *Let ϕ be a NMID-formula and I any structure interpreting all free symbols of ϕ. We define $I \models \phi$ (in words, ϕ* is true in *I, or I* satisfies *ϕ, or I is a* model *of ϕ) by the following induction:*

1. *$I \models X(t_1, .., t_n)$ if $(t^I_1, .., t^I_n) \in X^I$;*
2. *$I \models \Delta$ if $I = {I_{\mathrm{o}}}^{\Delta\downarrow} = {I_{\mathrm{o}}}^{\Delta\uparrow}$, where I_{o} is the restriction of I to τ^{o}_Δ;*
3. *$I \models \psi_1 \wedge \psi_2$ if $I \models \psi_1$ and $I \models \psi_2$;*
4. *$I \models \neg\psi$ if $I \not\models \psi$;*
5. *$I \models \exists\sigma\ \psi$ if for some value v of σ in the domain $dom(I)$ of I, $I[\sigma : v] \models \psi$.*

Given an NMID-theory T over τ, a τ-structure I satisfies *T (*is a model of *T) if I satisfies each $\phi \in T$. This is denoted by $I \models T$.*

The above inductive definition is a prototypcial example of a non-monotone inductive definition, more specifically a definition over a well-founded poset, namely the set of

[1] This operator is often called the Gelfond-Lifschitz operator and was introduced in (Gelfond & Lifschitz 1991).

NMID-formulas ordered by the subformula relation. It contains non-monotone recursion in rule 4. This is an example of a definition which can be represented using definitions in NMID-logic.

As mentioned before, the definitional implication should be distinguished from material implication. Rule $\forall \bar{x}\ (X(\bar{t}) \leftarrow \varphi)$ in a definition does not correspond to the disjunction $\forall \bar{x}\ (X(\bar{t}) \vee \neg\varphi)$, although it implies it. Intuitively, definitional implication should be understood as the "if" found in rules in inductive definitions (e.g. Definition 2 consists of 5 such rules).

Modularity Results

Definition 3 (partition of definitions). *A* partition of definition Δ *is a set* $\{\Delta_1, \ldots, \Delta_n\}$, $1 < n$, *such that* $\Delta = \Delta_1 \cup \cdots \cup \Delta_n$, *and if defined symbol* P *appears in the head of a rule of* Δ_i, $1 \leq i \leq n$, *then all rules of* Δ *with* P *in the head belong to* Δ_i *and only to* Δ_i.

Notice that Δ_i has some "new" open symbols. For instance, if P is defined in Δ, but not in Δ_i, then it is a new open symbol of Δ_i. Of course, it holds that $\tau = \tau^{\mathrm{o}}_{\Delta} \cup \tau^{\mathrm{d}}_{\Delta} = \tau^{\mathrm{o}}_{\Delta_i} \cup \tau^{\mathrm{d}}_{\Delta_i}$, $1 \leq i \leq n$. Also, $\cup_i \tau^{\mathrm{d}}_{\Delta_i} = \tau^{\mathrm{d}}_{\Delta}$ and $\tau^{\mathrm{d}}_{\Delta_i} \cap \tau^{\mathrm{d}}_{\Delta_j} = \emptyset$ whenever $i \neq j$.

A *domain atom* over vocabulary τ in domain A is any atom $P[a_1, \ldots, a_n]$ (or $P[\bar{a}]$), where P is relation symbol of τ and $a_1, \ldots, a_n$ are elements of A. Let At^{τ}_{A} be the set of domain atoms over τ in A. Let $\prec$ be any binary relation on At^{τ}_{A}. If $Q[\bar{b}] \prec P[\bar{a}]$, we will say that $P[\bar{a}]$ depends on $Q[\bar{b}]$ (according to $\prec$). We use $Q[\bar{b}] \prec\!\!\!\prec P[\bar{a}]$ as an abbreviation for $Q[\bar{b}] \prec P[\bar{a}] \wedge P[\bar{a}] \not\prec Q[\bar{b}]$. For any domain atom $P[\bar{a}]$ and any pair I, J of τ-structures with domain A, we define $I \cong_{\prec P[\bar{a}]} J$ if for each atom $Q[\bar{b}] \prec P[\bar{a}]$, $I \models Q[\bar{b}]$ iff $J \models Q[\bar{b}]$. We extend this to pairs by defining $(I, J) \cong_{\prec P[\bar{a}]} (I', J')$ if $I \cong_{\prec P[\bar{a}]} I'$ and $J \cong_{\prec P[\bar{a}]} J'$.

Let K_{o} be a structure with domain A interpreting at least all object and function symbols of τ and no defined predicates of Δ.

Definition 4 (reduction relation). *A binary relation* $\prec$ *on* At^{τ}_{A} *is a* reduction relation *(or briefly, a reduction) of* Δ *in* K_{o} *if for each domain atom* $P[\bar{a}]$ *with* P *a defined symbol, for all* τ*-structures* I, J, I', J' *extending* K_{o}, *if* $(I, J) \cong_{\prec P[\bar{a}]} (I', J')$ *then* $I_J \models \varphi_P[\bar{a}]$ *iff* $I'_{J'} \models \varphi_P[\bar{a}]$.

Intuitively, the definition expresses that $\prec$ is a reduction relation if the truth of the formulas $\varphi_P[\bar{a}]$ depends only on the truth of the atoms on which $P[\bar{a}]$ depends according to $\prec$.

Recall that a pre-well-founded order is a reflexive and transitive relation such that every non-empty subset contains a minimal element. The following definition is crucial for the Modularity theorem.

Definition 5 (reduction partition). *Call partition* $\{\Delta_1, \ldots, \Delta_n\}$ *of definition* Δ *a* reduction partition *of* Δ *in* $\tau^{\mathrm{o}}_{\Delta}$*-structure* I_{o} *if there is a reduction pre-well-founded order* $\prec$ *of* Δ *in* I_{o} *and if for each pair of atoms* $P[\bar{a}]$, $Q[\bar{b}]$ *which are not defined in the same* Δ_i *and such that* $Q[\bar{b}] \prec P[\bar{a}]$, *it holds that* $Q[\bar{b}] \prec\!\!\!\prec P[\bar{a}]$ *(i.e.,* $P[\bar{a}] \not\prec Q[\bar{b}]$*).*

The intuition underlying this definition is that in a reduction partition, if an atom defined in one module depends on an atom defined in another module, then the latter atom does not depend on the first atom and hence is strictly less in the reduction ordering.

A partition $\{\Delta_1, \ldots, \Delta_n\}$ of Δ is called total in K_{o} if each Δ_i is total in K_{o}.

Theorem 1 (modularity). *If* $\{\Delta_1, \ldots, \Delta_n\}$ *is a total reduction partition of* Δ *in* $\tau^{\mathrm{o}}_{\Delta}$*-structure* I_{o}, *then for any* τ*-structure* M *extending* I_{o}, $M \models \Delta_1 \wedge \cdots \wedge \Delta_n$ *iff* $M \models \Delta$.

Corollary 1. *Let* T_o *be a theory over* $\tau^{\mathrm{o}}_{\Delta}$ *such that for any* $\tau^{\mathrm{o}}_{\Delta}$*-model* M_o *of* T_o, $\{\Delta_1, \ldots, \Delta_n\}$ *is a total reduction partition of* Δ *in* M_o. *Then* $T_o \wedge \Delta$ *and* $T_o \wedge \Delta_1 \wedge \cdots \wedge \Delta_n$ *are logically equivalent.*

Now we consider for two special cases of definitions how to translate them in classical logic. Let Δ be a positive definition, i.e., with only positive occurrences of defined symbols in rule bodies, defining the symbols $\bar{P}$. Let X_i and P_i have the same arity. Define

$$PID(\Delta) := \bigwedge \Delta \wedge \forall \bar{X}(\bigwedge \Delta[\bar{P}/\bar{X}] \rightarrow \bar{P} \subseteq \bar{X}).$$

Here, $\bigwedge \Delta$ is the conjunction of formulas obtained by replacing definitional with material implications in Δ, $\Delta[\bar{P}/\bar{X}]$ is the definition obtained by substituting X_i for each defined symbol P_i and $\bar{P} \subseteq \bar{X}$ is a shorthand for the formula

$$\forall \bar{x}\ (P_1(\bar{x}) \rightarrow X_1(\bar{x})) \wedge \cdots \wedge \forall \bar{x}\ (P_n(\bar{x}) \rightarrow X_n(\bar{x})).$$

The formula $PID(\Delta)$ is the standard second-order formula to express that predicates $\bar{P}$ satisfy the positive inductive definition Δ.

Theorem 2. *If* Δ *is positive (i.e., contains no negative occurrences of defined symbols) then it is total in each* $\tau^{\mathrm{o}}_{\Delta}$*-structure, and for any structure* I, $I \models \Delta$ *iff* $I \models PID(\Delta)$.

The theory $PID(\Delta)$ expresses that the defined relations are the least relations closed under the rules of Δ in a $\tau^{\mathrm{o}}_{\Delta}$-structure. Because the rules of Δ are positive, such least relations are guaranteed to exist. Since these relations are the unique minimal relations closed under the rules of Δ, the well-known knowledge representation principle of circumscription could also be used to formalise Δ.

Another result is concerned with (possibly non-monotone) definitions over well-founded posets. First, we propose a formalisation for this informal concept in NMID-logic.

Definition 6 (strict reduction relation). *A reduction relation* $\prec$ *of* Δ *on* At^{τ}_{A} *is* strict *in* K_{o} *if it is a strict well-founded order (i.e., an antisymmetric, transitive binary relation without infinite descending chains).*

Thus, if $P[\bar{a}] \prec Q[\bar{b}]$ holds, then the bodies of rules defining $Q[\bar{b}]$ may depend on the truth value of $P[\bar{a}]$, but not vice versa.

Definition 7 (definition by well-founded induction). *Let* Δ *be a definition with a strict reduction relation* $\prec$ *in* K_{o}. *We call* Δ *a definition by well-founded induction (over* $\prec$*) in* K_{o}.

Being a definition by well-founded induction is not a universal property. A definition may have a strict reduction in one structure and not in other structures.

There is a well-known concept in knowledge representation that can be used to formalise this type of definitions in first-order logic. Define the *completion of* Δ, denoted $comp(\Delta)$, as the conjunction, for each defined symbol X of Δ, of formulas $\forall\bar{x}(X(\bar{x}) \leftrightarrow \varphi_X(\bar{x}))$.

Theorem 3 (completion). *Suppose Δ is a definition by well-founded induction in $\tau^{\mathrm{o}}_{\Delta}$-structure I_{o}. Then (a) definition Δ is total in I_{o}, and (b) the model ${I_{\mathrm{o}}}^{\Delta}$ of Δ is the unique model of $comp(\Delta)$ extending I_{o}.*

Corollary 2. *Let T_o be a theory over $\tau^{\mathrm{o}}_{\Delta}$ such that for any $\tau^{\mathrm{o}}_{\Delta}$-model M_o of T_o, Δ is a definition by well-founded induction in M_o. Then $T_o \wedge \Delta$ and $T_o \wedge comp(\Delta)$ are logically equivalent.*

Notice that positive inductive definitions and inductive definitions with strict reduction relation have different (and, in general, non-equivalent) formalisations in classical logic.

Reiter-style Situation Calculus

The vocabulary τ_{sc} of the situation calculus is a many-sorted vocabulary with equality and with sorts for actions (Act), situations (Sit), and possibly a finite number of domain-specific sorts called object sorts (Ob_1,..., Ob_k). The vocabulary contains a potentially infinite set of domain-dependent function symbols of the sort Act. The sort of each argument of such a function is an object sort. For example, in the block world domain, we may have actions $pick\_up(x)$ and $put\_on(x, y)$ ranging over the sort $Block$.

The vocabulary contains a binary relation $\sqsubseteq$ with arguments of sort Sit and denoting precedence of situations. The constant S_0 of sort Sit denotes the initial situation. Function do of sort Sit maps actions and situations to situations, i.e., given a and s, term $do(a, s)$ denotes the successor situation which is obtained from situation s by performing action a. The predicate constants $F_1, F_2, \ldots$ are called *fluents* and denote properties of the world (both in the initial situation and in other situations). Fluents always have exactly one argument of sort Sit, while the sort of each other argument is an object sort. For example, $On(x, y, s)$ of arity 3 denotes that object x is on object y in situation s.

Definition 8 ($\mathcal{D}_{\mathrm{una(S)}}$). *The theory of* unique name axioms for sort S, $\mathcal{D}_{\mathrm{una(S)}}$, *is the set of axioms in the following axiom schema: For distinct function symbols f and g of sort S*

$$\forall\bar{x}\,\forall\bar{y}\,\neg(f(\bar{x}) = g(\bar{y})). \tag{1}$$

$$\forall\bar{x}\,\forall\bar{y}\,(f(x_1,\ldots,x_n) = f(y_1,\ldots,y_n) \rightarrow x_1 = y_1 \wedge \cdots \wedge x_n = y_n). \tag{2}$$

The axioms (2) hold for every function symbol f with arity greater than zero.

Definition 9 ($\mathcal{D}_{\mathrm{f}}$). *The* foundational axioms of the situation calculus, $\mathcal{D}_{\mathrm{f}}$, *are the set of axioms consisting of the unique name axioms for situations $\mathcal{D}_{\mathrm{una(Sit)}}$, the domain closure axiom for situations*

$$\forall P\,(P(S_0) \wedge \forall s'\,\forall a\,(P(s') \rightarrow P(do(a, s'))) \rightarrow \forall s\,P(s)) \tag{3}$$

and the precedence axioms for situations

$$\forall s\,\neg(s \sqsubset S_0), \tag{4}$$

$$\forall s\,\forall s'\,\forall a\;\;(s \sqsubset do(a, s') \leftrightarrow s \sqsubseteq s') \tag{5}$$

where $s \sqsubset s'$ is an abbreviation for $s \sqsubseteq s' \wedge \neg(s' \sqsubseteq s)$.

The role of axiom (3) is to guarantee that the domain of situations Sit is the smallest set closed under applications of the function symbol do, which satisfies the unique name axioms for situations. Every two models of $\mathcal{D}_{\mathrm{f}}$ with identical domains of sort Act will have identical domains of sort Sit (modulo isomorphism).

Definition 10 ($\mathcal{D}_{\mathrm{ss}}$). *The* successor state axioms, $\mathcal{D}_{\mathrm{ss}}$, *are of the form:*

$$\forall\bar{x}\,\forall a\,\forall s\,(F(\bar{x}, do(a, s)) \leftrightarrow (\gamma^{+}_{F}(\bar{x}, a, s) \vee F(\bar{x}, s) \wedge \neg\gamma^{-}_{F}(\bar{x}, a, s))). \tag{6}$$

Formula $\gamma^{+}_{F}(\bar{x}, a, s)$ (respectively, $\gamma^{-}_{F}(\bar{x}, a, s)$) denotes a first-order formula specifying the conditions under which action a causes fluent F to become true (respectively, false) in the situation s (Reiter 1991). The only free variables of these formulas are among $\bar{x}$, a, s and the only symbol of sort Sit is the free variable s. An example of a successor axiom is

$$\forall sw\,\forall a\,\forall\, s\,(On(sw, do(a, s)) \leftrightarrow a = toggle(sw) \wedge \neg On(sw, s) \vee On(sw, s) \wedge a \neq toggle(sw)).$$

This axiom says that a switch is on in situation $do(a, s)$ if and only if this situation was obtained by performing action $toggle(sw)$ in situation s where this switch was off, or the switch was already on and an action other than $toggle(sw)$ was performed.

Definition 11 ($\mathcal{D}_{S_0}$). *A* description of the initial situation, $\mathcal{D}_{S_0}$, *is a set of first order sentences that are uniform in S_0, that is, it contains no situation term other than S_0.*

A *basic action theory* consists of $\mathcal{D}_{\mathrm{f}} \cup \mathcal{D}_{\mathrm{una(Act)}} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{\mathrm{ss}}$.

Inductive Situation Calculus

Here we define a variant of Reiter-style situation calculus, which we call the inductive situation calculus. All fluents will be defined by simultaneous induction on the well-founded set of situations. Ramifications describing propagation of effects of actions are modeled as monotone inductions at the level of situations. The result is an iterated inductive definition with alternating phases of monotone and non-monotone induction. Below we describe the components of the inductive situation calculus.

The vocabulary τ_{isc} of the inductive situation calculus extends τ_{sc} by two types of symbols. Symbols I_{F_1}, I_{F_2}, ... are used to describe the initial situation and correspond to the fluents $F_1, F_2, \ldots, F_n$ but have no situation argument. They are open symbols of the inductive situation calculus. The other type of symbols denote causality relations. These symbols will be introduced a bit later. The *open vocabulary* $\tau^{\mathrm{o}}_{\mathrm{isc}}$ of the inductive situation calculus consists of all symbols of τ_{isc} except for all fluents and causality predicates.

The *foundational axioms of the inductive situation calculus*, $\mathcal{D}_{\rm if}$, are the unique name axioms $\mathcal{D}_{\rm una(Sit)}$ for situations, the following domain closure axiom for situations

$$\exists P \left[\left\{ \begin{array}{l} \forall s\ (P(s) \leftarrow s = S_0), \\ \forall a \forall s\ (P(do(a,s)) \leftarrow P(s)) \end{array} \right\} \wedge \forall s P(s) \right] \quad (7)$$

and the following definition of the precedence relation

$$\Delta_{\sqsubseteq} := \left\{ \begin{array}{l} \forall s \forall s'\ (s \sqsubseteq s' \leftarrow s = s'), \\ \forall s \forall s' \forall a\ (s \sqsubseteq do(a,s') \leftarrow s \sqsubseteq s') \end{array} \right\}. \quad (8)$$

The role of axiom (7) is to guarantee that the domain of situations Sit is the smallest set closed under applications of the function symbol do, which satisfies the unique name axioms for situations. It is equivalent to Reiter's induction axiom for situations.

In place of $\mathcal{D}_{S_0}$, the description of the initial situation in terms of fluents which hold in S_0, in the inductive situation calculus we describe the initial situation in terms of symbols I_{F_i}. The corresponding collection of axioms is $\mathcal{D}_{\rm init}$.

An *initial structure* $\mathcal{A}_{\rm I}$ of the inductive situation calculus is a multi-sorted structure with a non-empty domain for each sort of the language, which interprets all symbols of $\tau^{\rm o}_{\rm isc}$ and which satisfies the foundational axioms $\mathcal{D}_{\rm if}$ and the unique name axioms for actions $\mathcal{D}_{\rm una(Act)}$.

Proposition 1. *Let $\mathcal{A}$ be a $\tau^{\rm o}_{\rm isc}$-structure, which satisfies $\mathcal{D}_{\rm if}$ and the definition $\Delta_{\sqsubseteq}$ (8). In every such structure, $\langle Sit^{\mathcal{A}}, \sqsubseteq^{\mathcal{A}} \rangle$ is a well-founded poset (and, thus a pre-well-founded set).*

The following proposition demonstrates that the remaining two foundational axioms of the situation calculus as presented in (Reiter 2001), are implied by the definition above.

Proposition 2. *The sentences (4) and (5) are logically implied by the unique name assumptions for situations $\mathcal{D}_{\rm una(Sit)}$, the domain closure axiom for situations (7), and by the sentence (8).*

Proof. The definition $\Delta_{\sqsubseteq}$ logically implies its completion $comp(\Delta_{\sqsubseteq})$. This theory gives us $\forall s\ \forall s'\ (s \sqsubseteq s' \leftrightarrow s = s' \vee \exists a\ \exists s''\ \ s' = do(a,s'') \wedge s \sqsubseteq s'')$. To prove (4), we instantiate the completion with $s' = S_0$, taking into account the unique name assumptions for situations. We obtain: $\forall s\ \ (s \sqsubseteq S_0 \leftrightarrow s = S_0)$. This sentence implies $\forall s\ \ (s \sqsubseteq S_0 \rightarrow s = S_0)$, which entails $\forall s\ \neg(s \sqsubset S_0)$.

To prove (5), we use a model theoretic argument. In each structure satisfying the foundational axioms, there is a bijection (one to one and onto) from the set of finite sequences of elements of the action sort to the situation sort. By a simple inductive argument on the definition of $\sqsubseteq$, one can show that for two situations s, s', $s \sqsubseteq s'$ if and only if the sequence of actions of s is an initial segment of the sequence of actions of s'. When $s \sqsubset do(a,s')$ holds, then the action sequence in s is a strictly smaller initial segment of the action sequence of $do(a,s')$ and hence an initial segment of the action sequence of s'. Consequently, $s \subseteq s'$. Vice versa, if $s \sqsubseteq s'$, then the action sequence of s will be a strictly smaller initial segment of the action sequence of s' appended with action a, and hence $s \sqsubset do(a,s')$ holds. □

A *basic action theory of the inductive situation calculus* will be a collection of axioms of the form:

$$\mathcal{D}_{\rm if} \cup \mathcal{D}_{\rm una(Act)} \cup \mathcal{D}_{\rm init} \cup \{\Delta_{\rm sc}\}, \quad (9)$$

where $\Delta_{\rm sc}$ is an inductive definition of the fluents. In the next sections, we present two variants of $\Delta_{\rm sc}$.

Specifying Direct Effects of Actions For each fluent F_i, we introduce two additional auxiliary relations, C_{F_i} and $C_{\neg F_i}$. These relations represent initiating and terminating causes for F_i, respectively. Both C_{F_i} and $C_{\neg F_i}$ have the same sort of arguments as F_i plus one action argument. Let $\mathcal{D}_{\rm init}$ axiomatize the initial situation using $I_{F_1}, \ldots I_{F_m}$.

We augment $\mathcal{D}_{\rm if} \cup \mathcal{D}_{\rm una(Act)} \cup \mathcal{D}_{\rm init}$ with the following definition

$$\Delta_{\rm sc} = \bigcup_i \Delta^i_{\rm fluent} \cup \bigcup_i \Delta^i_{\rm effect}$$

where $\Delta^i_{\rm fluent} :=$

$$\left\{ \begin{array}{rcl} \forall \bar{x}_i\ (F(\bar{x}_i, S_0) & \leftarrow & I_F(\bar{x}_i)), \\ \forall \bar{x}_i\ (F_i(\bar{x}_i, do(a,s)) & \leftarrow & C_{F_i}(\bar{x}_i, a, s)), \\ \forall \bar{x}_i\ (F_i(\bar{x}_i, do(a,s)) & \leftarrow & F_i(\bar{x}_i, s) \wedge \\ & & \neg C_{\neg F_i}(\bar{x}_i, a, s)) \end{array} \right\},$$

and $\Delta^i_{\rm effect} :=$

$$\left\{ \begin{array}{rcl} \forall \bar{x}_i\ (C_{F_i}(\bar{x}_i, a, s) & \leftarrow & \gamma^+_{F_i}(\bar{x}_i, a, s)), \\ \forall \bar{x}_i\ (C_{\neg F_i}(\bar{x}_i, a, s) & \leftarrow & \gamma^-_{F_i}(\bar{x}_i, a, s)) \end{array} \right\}.$$

The intuitive meaning of this definition is as follows. The first rule of $\Delta^i_{\rm fluent}$ defines the fluent in situation S_0. The second rule says that if an action causes a fluent in some situation, then the fluent holds in the successor situation. The third rule deals with the case where a fluent is not affected by an action and will be referred to as the *law of inertia*. The rules in $\Delta^i_{\rm effect}$ describe direct effects of actions on the fluent F_i. Formulas $\gamma^+_{F_i}(\bar{x}_i, a, s)$, $\gamma^-_{F_i}(\bar{x}_i, a, s)$ are analogous to those found in Reiter-style situation calculus. The only situation term appearing in them is s and they do not contain causality or initiation predicates. From the formulas $\gamma^+_{F_i}(\bar{x}_i, a, s)$, $\gamma^-_{F_i}(\bar{x}_i, a, s)$, a successor state axiom for the fluent F_i can be constructed. These axioms will be called the successor state axioms corresponding to $\Delta_{\rm sc}$.

Note that any fluent may appear in the rules of a causality predicate. Hence, this definition is one large simultaneous inductive definition. Moreover, since the inertia law contains a negative occurrence of $C_{\neg F_i}$ and this predicate may be defined in terms of fluents, this is, in general, a non-monotone inductive definition.

In what follows, we use our modularity results in order to obtain the standard successor state axioms of the situation calculus.

Proposition 3. *Suppose $\mathcal{A}_{\rm I}$ is an initial structure of the inductive situation calculus such that $\mathcal{A}_{\rm I} \models \mathcal{D}_{\rm init}$. A structure I extending $\mathcal{A}_{\rm I}$ is a model of $\Delta_{\rm sc}$ iff I is a model of $comp(\Delta_{\rm sc})$.*

Proof. Let A be the domain of $\mathcal{A}_{\rm I}$. Define the following relation $\prec$ on $At^{\mathcal{A}_{\rm I}}_{\tau_{isc}}$, the set of all domain atom, as the set of

all tuples:

$$(I_{P_i}[\bar{u}],\ P_i[\bar{u}, S_0^{\mathcal{A}_\mathrm{I}}]),$$
$$(C_{(\neg)P_i}[\bar{u}, a, s],\ P_i[\bar{u}, do^{\mathcal{A}_\mathrm{I}}(a, s)]),$$
$$(P_i[\bar{u}, s],\ C_{(\neg)P_j}[\bar{v}, a, s]),$$

for arbitrary tuples of objects $\bar{u}$ and $\bar{v}$, for arbitrary elements a of the action sort and s of the situation sort, for each i, j.

It is easy to show that $\prec$ is a reduction relation. Since any superset of a reduction relation is also a reduction relation, the transitive closure $\prec^*$ of $\prec$ is a reduction relation. Moreover, it follows from the fact that $\langle Sit^{\mathcal{A}_\mathrm{I}}, \sqsubseteq^{\mathcal{A}_\mathrm{I}} \rangle$ is a pre-well-founded set (Proposition 1), that $\prec^*$ is a strict pre-well-founded order on $At^{\mathcal{A}_\mathrm{I}}_{\Delta_\mathrm{sc}}$. This means that we can apply Theorem 3 and we obtain that Δ_sc is total in $\mathcal{A}_\mathrm{I}$ and is equivalent to the completion. □

Suppose that τ_1, τ_2 are vocabularies extending τ and let T_1, T_2 be theories in respectively τ_1, τ_2. We call T_1 equivalent in τ to T_2 if for each τ_1-model M_1 of T_1, there exists a τ_2-model M_2 of T_2 such that $M_1|_\tau = M_2|_\tau$ and vice versa. Here $M_i|_\tau$ denotes the restriction of M_i to the symbols of τ.

Proposition 4. *$\mathcal{D}_\mathrm{if} \cup \mathcal{D}_{\mathrm{una(Act)}} \cup \mathcal{D}_\mathrm{init} \cup \{\Delta_\mathrm{sc}\}$ is equivalent in τ_{sc} to*

$$\mathcal{D}_\mathrm{f} \cup \mathcal{D}_{\mathrm{una(Act)}} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_\mathrm{ss}$$

where $\mathcal{D}_{S_0}$ is the theory obtained from $\mathcal{D}_\mathrm{init}$ by substituting $F_i(\bar{t}, S_0)$ for each atom $I_{F_i}(\bar{t})$ and $\mathcal{D}_\mathrm{ss}$ is the set of the successor state axioms corresponding to Δ_sc.

Proof. $\mathcal{D}_\mathrm{if}$ and $\mathcal{D}_\mathrm{f}$ are logically equivalent. By proposition 3, $\mathcal{D}_\mathrm{f} \cup \mathcal{D}_\mathrm{init} \cup \{\Delta_\mathrm{sc}\}$ is logically equivalent to $\mathcal{D}_\mathrm{f} \cup \mathcal{D}_\mathrm{init} \cup comp(\Delta_\mathrm{sc})$, where $comp(\Delta_\mathrm{sc})$ is

$$\begin{array}{l}
\bigwedge_i^n \forall \bar{x}_i\ F_i(\bar{x}_i, s) \leftrightarrow (s = S_0 \wedge I_{P_i}(\bar{x}_i) \vee \\
\quad \exists a\ \exists s' \quad s = do(a, s') \wedge \\
\qquad\qquad C_{F_i}(\bar{x}_i, s) \vee F_i(\bar{x}_i, s') \wedge \neg C_{\neg F_i}(\bar{x}_i, s)) \\
\wedge \bigwedge_i^n \forall \bar{x}_i \forall s \forall a\ C_{F_i}(\bar{x}_i, a, s) \leftrightarrow \gamma^+_{F_i}(\bar{x}_i, a, s') \\
\wedge \bigwedge_i^n \forall \bar{x}_i \forall s \forall a\ C_{\neg F_i}(\bar{x}_i, a, s) \leftrightarrow \gamma^-_{F_i}(\bar{x}_i, a, s').
\end{array} \tag{10}$$

Since, by the domain closure axiom for situations,

$$\forall s\ s = S_0 \vee \exists a\ \exists s'\ s = do(a, s'),$$

$\mathcal{D}_\mathrm{f} \cup \{(10)\}$ is logically equivalent to $\mathcal{D}_\mathrm{f} \cup \{(11), (12)\}$, where

$$\begin{array}{l}
\bigwedge_i^n \forall \bar{x}_i \forall s \forall a\ F_i(\bar{x}_i, do(a, s)) \leftrightarrow \\
\qquad C_{F_i}(\bar{x}_i, do(a, s)) \vee \\
\qquad F_i(\bar{x}_i, s) \wedge \neg C_{\neg F_i}(\bar{x}_i, do(a, s))
\end{array} \tag{11}$$

and

$$\begin{array}{l}
\bigwedge_i^n \forall \bar{x}_i\ F_i(\bar{x}_i, S_0) \leftrightarrow I_{P_i}(\bar{x}_i) \\
\wedge \bigwedge_i^n \forall \bar{x}_i \forall s \forall a\ C_{F_i}(\bar{x}_i, a, s) \leftrightarrow \gamma^+_{F_i}(\bar{x}_i, a, s) \\
\wedge \bigwedge_i^n \forall \bar{x}_i \forall s \forall a\ C_{\neg F_i}(\bar{x}_i, a, s) \leftrightarrow \gamma^-_{F_i}(\bar{x}_i, a, s).
\end{array} \tag{12}$$

Given the equivalences in (12), it is clear that $\mathcal{D}_\mathrm{if} \cup \mathcal{D}_{\mathrm{una(Act)}} \cup \mathcal{D}_\mathrm{init} \cup \{(11), (12)\}$ is logically equivalent to $\mathcal{D}_\mathrm{f} \cup \mathcal{D}_{\mathrm{una(Act)}} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_\mathrm{ss} \cup \{(12)\}$.

Finally, observe that in the latter theory, the predicate symbols I_{P_i}, C_{F_i} and $C_{\neg F_i}$ occur only at the lefthand-side of the explicit definitions in (12). It follows that $\mathcal{D}_\mathrm{f} \cup \mathcal{D}_{\mathrm{una(Act)}} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_\mathrm{ss} \cup \{(12)\}$ is equivalent in τ_sc to $\mathcal{D}_\mathrm{f} \cup \mathcal{D}_{\mathrm{una(Act)}} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_\mathrm{ss}$.

□

Note that our definition Δ^i_fluent does not contain rules of the form

$$\forall \bar{x}_i \forall a \forall s\ (\neg P_i(\bar{x}_i, a, s) \leftarrow C_{\neg P_i}(\bar{x}_i, a, s)). \tag{13}$$

However, under a natural requirement, we can derive negative effect axioms of actions, as we demonstrate below. The requirement is that a fluent and its negation are not caused to hold in the same situation. Formally, the requirement is that the basic action theory should entail the following sentence :

$$\bigwedge_i^n \forall \bar{x}_i \forall a \forall s\ \neg(\gamma^+_{F_i}(\bar{x}_i, a, s) \wedge \gamma^-_{F_i}(\bar{x}_i, a, s)).$$

It is easy to show now that if this requirement is satisfied, then the negative effect axiom is implied. Observe that each successor state axiom entails

$$\begin{array}{l}
\forall \bar{x}_i \forall a \forall s\ \neg\gamma^+_{F_i}(\bar{x}_i, do(a, s)) \wedge \gamma^-_{F_i}(\bar{x}_i, do(a, s)) \\
\qquad \rightarrow \neg F_i(\bar{x}_i, do(a, s)).
\end{array}$$

Under the requirement, the first literal in the condition is entailed by the second, so we can drop it and we obtain the negative effect rule

$$\forall \bar{x}_i \forall a \forall s\ (\gamma^-_{F_i}(\bar{x}_i, do(a, s)) \rightarrow \neg F_i(\bar{x}_i, do(a, s))).$$

Therefore, in the context of Inductive Situation Calculus, rule (13) is not necessary. This observation illustrates a general principle of inductive definitions. In an inductive definition, one defines a concept by enumerating positive cases. Given such an enumeration, the closure mechanism underlying inductive definitions yields the negative cases.

Indirect Effects The ramification problem arises in the context of knowledge representation, when one wants to capture indirect effects of actions in a logic-based formalism. It has been shown (e.g., (Lin 1995)) that state constraints are generally inadequate for deriving indirect effects of actions, and that some notion of causation is needed. Unlike material implication, causal implications are not contrapositive which makes them similar to the rules of inductive definitions. This property is the foundation of our solution to the ramification problem. The semantic correspondence between causality rules and rules in an inductive definition was independently pointed out in (Ternovskaia 1998a; 1998b) and in (Denecker, Theseider Duprè, & Van Belleghem 1998).

Let, as before, C_{F_i} and $C_{\neg F_i}$ represent initiating and terminating causes for F_i, respectively. We extend the use of the causality predicates to specify indirect effects of actions. For example, according to the causal rule $\forall a \forall s\ (C_{F_2}(a, s) \leftarrow C_{\neg F_1}(a, s))$, when an action a causes termination of F_1, then the same action, indirectly, causes the initiation of F_2. We relax the conditions on Δ^i_effect, so that any number of rules of the following form can appear in it:

$$\begin{array}{lll}
\forall a \forall s\ (C_{F_i}(\bar{x}_i, a, s) & \leftarrow & \Psi^+_{F_i}(\bar{x}_i, a, s)), \\
\forall a \forall s\ (C_{\neg F_i}(\bar{x}_i, a, s) & \leftarrow & \Psi^-_{F_i}(\bar{x}_i, a, s))
\end{array}, \tag{14}$$

where Ψ^+ and Ψ^- are formulas in which s is the only situation term. Note that in the direct effect case, causality predicates were excluded from bodies of rules of Δ^i_effect.

The basic action theory (9), where $\Delta_{\rm sc}$ is as above, encodes our *most general solution to the ramification problem* in the inductive situation calculus.

Consider the following partition of $\Delta_{\rm sc}$

$$\{\ \Delta^1_{\rm effect} \cup \cdots \cup \Delta^n_{\rm effect},\quad \Delta^1_{\rm fluent}, \ldots, \Delta^n_{\rm fluent}\ \} \qquad (15)$$

Proposition 5. *Suppose $\mathcal{A}_{\rm I}$ is any initial structure of the inductive situation calculus and causality predicates have only positive occurrences in $\Delta_{\rm sc}$. Partition (15) is a total reduction partition of $\Delta_{\rm sc}$ in $\mathcal{A}_{\rm I}$.*

Proof. Let A be the domain of $\mathcal{A}_{\rm I}$. Define the following relation $\prec$ on $At^{\mathcal{A}_{\rm I}}_{\tau_{isc}}$, the set of all domain atoms, as the set of all tuples:

$$\begin{array}{l}(\ I_{P_i}[\bar{u}],\ P_j[\bar{u}, S_0^{\mathcal{A}_{\rm I}}]\ ),\\ (\ C_{(\neg)P_i}[\bar{u}, a, s],\ P_j[\bar{u}, do^{\mathcal{A}_{\rm I}}(a, s)]\ ),\\ (\ P_i[\bar{u}, s],\ C_{(\neg)P_j}[\bar{v}, a, s]\ ),\\ (\ C_{(\neg)P_i}[\bar{u}, a, s],\ C_{(\neg)P_j}[\bar{v}, a, s]\ ),\end{array}$$

for arbitrary tuples of objects $\bar{u}$ and $\bar{v}$, for arbitary elements a of the action sort and s of the situation sort and for each i, j.

It is easy to show that $\prec$ is a reduction relation. Since any superset of a reduction relation is also a reduction relation, the reflexive, transitive closure $\prec^*$ is a reduction relation. Moreover, it follows from the fact that $\langle Sit^{\mathcal{A}_{\rm I}}, \sqsubseteq^{\mathcal{A}_{\rm I}}\rangle$ is a pre-well-founded set (Proposition 1), that $\prec^*$ is a pre-well-founded order on $At^{\mathcal{A}_{\rm I}}_{\Delta_{\rm sc}}$. It is easy to see that for atoms $P[\bar{a}]$, $Q[\bar{b}]$ from $At^{\mathcal{A}_{\rm I}}_{\Delta_{\rm sc}}$, if $Q[\bar{b}] \prec^* P[\bar{a}]$ and $P[\bar{a}]$, $Q[\bar{b}]$ are not defined in the same Δ_i then $Q[\bar{b}] \prec\!\!\prec^* P[\bar{a}]$. Therefore, partition (15) is a reduction partition of $\Delta_{\rm sc}$.

Observe that each definition in partition (15) is positive, and, therefore total in each structure. Consequently, partition (15) is a total reduction partition in $\mathcal{A}_{\rm I}$.

□

Theorem 4. *If causality predicates have only positive occurrences in $\Delta_{\rm sc}$ then the basic action theory (9) is equivalent to the theory $\mathcal{D}_{\rm f} \cup \mathcal{D}_{\rm una(Act)} \cup \mathcal{D}_{S_0} \cup \{\bigwedge_i comp(\Delta^i_{\rm fluent}) \wedge PID(\bigcup_i \Delta^i_{\rm effect})$.*

Proof. By proposition 5, we have a total reduction partition. By the modularity theorem 1, we can split the definition. The definition $\bigcup_i \Delta^i_{\rm effect}$ is a positive definition which, by theorem 2, can be translated into $PID(\bigwedge_i \Delta^i_{\rm effect})$. The definitions $\Delta^i_{\rm fluent}$ have strict reduction relations, so they can be transformed into completions by theorem 3.

□

Example: N Gear Wheels Let us describe a simple idealized mechanical system consisting of a number of gear wheels $w_1, \ldots, w_n$, each pair of which may or may not be mechanically connected. For each of these wheels, we consider two states: *turning* or *stopped*. For each of these wheels, we consider two actions $start(w_i)$ and $stop(w_i)$. The first action gives an impulse to the wheel which propagates over the system to all connected gearwheels; the second action brakes the wheel and all connected wheels. We assume that once a wheel turns, it continues to turn (there is no friction; this system behaves as a perpetuum mobile) until there is a stop action.

We are faced here with a ramification problem — the problem of how to describe the propagation of effects through the system of connected gear wheels. The goal is to develop a *modular* temporal theory describing the effects of the basic actions and the propagation of effects. As a correctness criterion, we should be able to prove the state constraint that in all situations, a gear wheel w is turning if and only if all reachable wheels (those connected to w in the transitive closure of the connection graph) are turning as well.

We could represent this example in Reiter's basic situation calculus (Reiter 2001). To do this we could pre-compute for each wheel the set of reachable wheels in the connection graph; it suffices then to express that the action of starting (resp. stopping) a wheel w has the immediate effect to initiate (resp. terminate) the turning state of wheel w and each wheel reachable from w. This representation would have an important drawback due to the fact that it contains an explicit representation of the transitive closure of the physical connections between gear wheels. This relation is an example of a *global property* of the system which emerges as an interaction of *local properties*, namely the physical connections between gear wheels. If we explicitly represent such global properties then a small change of a local property (e.g. adding a new connection or deleting an existing connection between two gear wheels) may have a strong impact on the global properties and hence on the theory (e.g. disconnecting one pair of gear wheels may split a large interconnected set of connected wheels and would affect the representation of the effect of all actions on all wheels in this set). In a *modular* representation, only local properties of the components should be represented explicitly; global properties should be derivable from a generic part of the theory which does not explicitly depend on the actual configuration of the system. This is an aspect of *elaboration tolerance* (McCarthy 1998).

To obtain a modular representation in the gear wheel example, we need to be able to express the reachability from a specific wheel in an arbitrary graph. It is wel-known that this concept cannot be expressed in first order logic. Below we present a formalisation through an iterated inductive definition.

In the gear wheel example, there is one domain dependent sort, denoted $Gearwheel$. Action symbols are $start$ and $stop$ and have sort $\langle Gearwheel\rangle$. The unique fluent $Turns$ has sort $\langle Gearwheel, Sit\rangle$.

Basic components of the inductive situation calculus for the Gearwheel example are the foundational axioms $\mathcal{D}_{\rm if}$ of situations and the unique name axioms $\mathcal{D}_{\rm una(Act)}$ for actions. The main axiom of our theory is the simultaneous iterated inductive definition $\Delta_{\rm sc}$ of the fluent $Turns$ and its causality predicates C_{Turns} and $C_{\neg Turns}$. The effect propagation process caused by start or stop actions in one situation will be modeled by a monotone induction. To define the fluent $Turns$ for all states, the monotone induction is then iterated over the well-founded structure of situations.

The definition $\Delta_{\rm sc}$ can be split up in two subdefinitions.

The first part of the definition consists of the rules for $Turns$: $\Delta_{Turns} :=$

$$\left\{ \begin{array}{l} \forall s \forall g\, Turns(g, S_0) \leftarrow I_{Turns}(g) \\ \forall a \forall s \forall g\, Turns(g, do(a, s)) \leftarrow C_{Turns}(g, a, s) \\ \forall a \forall s \forall g\, Turns(g, do(a, s)) \leftarrow \\ \qquad\qquad Turns(g, s) \wedge \\ \qquad\qquad \neg C_{\neg Turns}(g, a, s) \end{array} \right\}$$

Notice that the third rule, the law of inertia, contains recursion over negation.

The second part of the definition Δ_{effect} describes the causation predicates C_{Turns} and $C_{\neg Turns}$. The following set of rules Δ_{effect} specify direct and indirect effects of actions:

$$\left\{ \begin{array}{l} \forall a \forall s \forall g\, C_{Turns}(g, a, s) \leftarrow a = Start(g), \\ \forall a \forall s \forall g\, C_{\neg Turns}(g, a, s) \leftarrow a = Stop(g)), \\ \forall a \forall s \forall g\, C_{Turns}(g, a, s) \leftarrow \\ \qquad\qquad \exists g'\ \ Connected(g, g') \wedge \\ \qquad\qquad\quad\ C_{Turns}(g', a, s), \\ \forall a \forall s \forall g\, C_{\neg Turns}(g, a, s) \leftarrow \\ \qquad\qquad \exists g'\ \ Connected(g, g') \wedge \\ \qquad\qquad\quad\ C_{\neg Turns}(g', a, s) \end{array} \right\}$$

These rules contain positive recursion. To represent the physical connections between the gear wheels, we used the binary relation symbol predicate $Connected$. This is a symmetric relation, as is expressed by the theory $\mathcal{D}_{Conn}$ consisting of the axiom:

$$\forall g \forall g'\, Connected(g, g') \rightarrow Connected(g', g).$$

Define $\Delta_{\text{sc}} := \Delta_{Turns} \cup \Delta_{effect}$. This definition defines the predicates $Turns$, C_{Turns} and $C_{\neg Turns}$ by simultaneously non-monotone induction in terms of the open predicates I_{Turns} and $Connected$.

We assume that in the initial state, all gear wheels are in rest. This is expressed by the theory $\mathcal{D}_{\text{init}}$ which contains one axiom

$$\forall g\, \neg I_{Turns}(g).$$

The full axiomatisation of the domain consists of

$$\mathcal{D}_{wheels} := \mathcal{D}_{\text{if}} \cup \mathcal{D}_{\text{una(Act)}} \cup \mathcal{D}_{Conn} \cup \mathcal{D}_{\text{init}} \cup \{\Delta_{\text{sc}}\}.$$

Notice that the configuration of the gear wheels is left unspecified both in the statement of the problem and in its axiomatisation $\mathcal{D}_{wheels}$.

Below we analyse the theory $\mathcal{D}_{wheels}$. Since Δ_{effect} is a positive definition, the basic action theory $\mathcal{D}_{wheels}$ satisfies the conditions of theorem 4. Consequently, we have the following proposition.

Proposition 6. *The theory $\mathcal{D}_{wheels}$ is equivalent to*

$$\begin{array}{c} \mathcal{D}_{\text{if}} \cup \mathcal{D}_{\text{una(Act)}} \cup \mathcal{D}_{Conn} \cup \mathcal{D}_{\text{init}} \cup \\ PID(\Delta_{\text{Effect}})\} \cup comp(\Delta_{\text{Turns}}) \end{array}$$

Proposition 7. *In each situation, all connected gear wheels are turning or they are all in rest. The theory $\mathcal{D}_{wheels}$ logically entails:*

$$\exists X \left(\begin{array}{l} \left\{ \begin{array}{l} \forall g \forall g'\, X(g, g') \leftarrow Connected(g, g'), \\ \forall g \forall g'\, X(g, g') \leftarrow \exists g''\ \ X(g, g'') \wedge \\ \qquad\qquad\qquad\qquad\ X(g'', g') \end{array} \right\} \\ \wedge \forall s \forall g \forall g'\, (X(g, g') \rightarrow \\ \qquad (Turns(g, s) \leftrightarrow Turns(g', s))) \end{array} \right)$$

Proof. The proof is model theoretic. Let I be a model of $\mathcal{D}_{wheels}$.

The proof is by induction on the length of the situations. Since all gear wheels are in rest in the initial situation, the property is trivially satisfied in this situation. Assume that the property is satisfied for situation s. We prove that it holds for the successor situation $do^I(a, s)$, for arbitrary action a.

Assume that there is a path from gearwheel g to gearwheel g' through the graph $Connected^I$. By definition Δ_{effect}, if $C_{Turn}(g', a, s)$ is true then so is $C_{Turn}(g, a, s)$. Because the graph $Connected^I$ is symmetric, it follows that $C_{Turn}(g, a, s)$ and $C_{Turn}(g', a, s)$ have the same truth value. The same holds for $C_{\neg Turn}$. The induction hypothesis states that in situation s all connected wheels are in the same state. By the above observation, the action a has the same effects on all connected wheels. Consequently, the induction hypothesis is preserved in situation $do^I(a, s)$. □

Related work For an overview of different approaches for temporal reasoning and the ramification problem we refer to (Thielscher 1997) (McIlraith 2000) (Denecker, Theseider Duprè, & Van Belleghem 1998). Here we limit our discussion to approaches based on situation calculus using inductive definitions and classical logic.

The idea of using inductive definitions for modeling temporal reasoning using inductive definitions was pointed out independently in (Ternovskaia 1998a; 1998b) and (Denecker, Theseider Duprè, & Van Belleghem 1998). In both cases, the motivation for using inductive definitions was the similarity between the process of effect propagation in a dynamic system and inductive definitions. Basically, the process of effect propagation is a *constructive* process: basic actions cause changes and effects which propagate through the dynamic system; changes do not appear without an external cause. The same constructive intuition is found in inductive definitions. This explains why inductive definitions can correctly model recursive effect propagations. In this respect, the inductive situation calculus is more general than two other well-known classical logic formalisations of the situation calculus with ramifications, namely Lin's approach (Lin 1995) and McIlraith's solitary stratified theories (McIlraith 2000). Both approaches impose acyclicity constraints on ramification rules which preclude recursive ramifications. A strong constraint in solitary stratified theories is that no fluent symbol is allowed to appear both as an effect and in the precondition of the same action. On the other hand, McIlraith addresses the qualification problem, which we don't.

Conclusion

This paper explains the inductive nature of situation calculus. We have shown that — unsuspected by its creators — the original Reiter-style situation calculus and its extension for representing ramification, makes hidden use of inductive definitions. We made these definitions explicit and found monotone and non-monotone induction. In the Reiter-style situation calculus, these different forms of induction are formalised in different ways. In our representation in NMID-

logic, they can be represented in a uniform way. In this sense our representation is simpler and may lead to a more modular representation. We presented a translation to classical logic to show that our formalisation of situation calculus is indeed equivalent to the standard formalisation. Finally, our experiment also demonstrates that the use of different forms of inductive definitions is not limited to mathematics, but may have applications in a much wider area of knowledge representation, including commonsense reasoning.

References

Denecker, M., and Ternovska, E. 2004. A logic for non-monotone inductive definitions and its modularity properties. In *Proc. of LPNMR-04*.

Denecker, M.; Bruynooghe, M.; and Marek, V. 2001. Logic programming revisited: Logic programs as inductive definitions. *ACM Transactions on Computational Logic (TOCL)* 4(2).

Denecker, M.; Theseider Duprè, D.; and Van Belleghem, K. 1998. An inductive definition approach to ramifications. *Linköping Electronic Articles in Computer and Information Science* 3(1998): nr 7. URL: http://www.ep.liu.se/ea/cis/1998/007/.

Denecker, M. 2000. Extending classical logic with inductive definitions. In *Proc. CL'2000*.

Fitting, M. 2003. Fixpoint semantics for logic programming - a survey. *Theoretical Computer Science*. To appear.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385.

Lin, F. 1995. Embracing causality in specifying the indirect effects of actions. In *Proc. of IJCAI 95*, 1985–1991.

McCarthy, J. 1998. Elaboration tolerance. In *COMMON SENSE 98, Symposium On Logical Formalizations Of Commonsense Reasoning*.

McIlraith, S. 2000. An axiomatic solution to the ramification problem (sometimes). *Artificial Intelligence* 116(1-2):87–121.

Reiter, R. 1991. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*. San Diego, CA: Academic Press. 359–380.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. MIT Press.

Ternovskaia, E. 1998a. Causality via inductive definitions. In *Working Notes of "Prospects for a Commonsense Theory of Causation", AAAI Spring Symposium Series, March 23-28*.

Ternovskaia, E. 1998b. Inductive definability and the situation calculus. In *Transaction and Change in Logic Databases*, volume 1472 of *Lecture Notes in Computer Science*. Springer-Verlag.

Thielscher, M. 1997. Ramification and causality. *J. of Artificial Intelligence* 89:317–364.

Van Gelder, A. 1993. An alternating fixpoint of logic programs with negation. *Journal of computer and system sciences* 47:185–221.