Putting $\mathcal{E}$ into classical logic

Tom Costello

Kakas and Miller suggest an action language, $\mathcal{E}$, which they extend to deal with ramifications. We show how their language can naturally be embedded in a classical language. Action languages have become popular, because they are claimed to be *easily readable, understandable and intuitive*. Some people (especially the author) find it more natural to write in classical logic. Thus this note transcribes the notation of Kakas and Miller to classical second order logic.

We first present a family of second order languages, and then show that this family captures (the finite part of) the $\mathcal{E}$ family.

Thus, rather than use a new class of languages, we use second order logic, with three non-empty sorts, actions a, fluents f, and times t, using g for sets of fluents, with predicates, $t \leq t'$, $holds(f,t)$, $happens(a,t)$, $initiates(a,f,g)$, and $terminates(a,f,g)$. We also add the predicate $whenever(f,g)$ later. We have a function not on fluents, which is of period two, and interacts with the $holds$ predicate [1] on fluents in the natural way.

$$\begin{array}{l} \forall f.not(not(f)) = f \\ \forall f,f'.not(f) = not(f') \rightarrow f = f' \\ \neg(F = not(F')), \quad \text{for constants } F, F' \\ \forall f,t.holds(f,t) \equiv \neg holds(not(f),t) \end{array} \tag{1}$$

We postulate partial order axioms for $\leq$,

$$\begin{array}{l} \forall t.t \leq t' \\ \forall t,t',t''.t \leq t' \wedge t' \leq t'' \rightarrow t \leq t'' \\ \forall t,t'.t \leq t' \wedge t' \leq t \rightarrow t = t' \end{array} \tag{2}$$

We define predicates $term(f,t)$ and $init(f,t)$, which captures the notion of t being an initiation point/ termination point for f.

$$\forall f,t.init(f,t) \equiv \exists a,g.happens(a,t) \wedge initiates(a,f,g) \wedge (\forall f'.g(f') \rightarrow holds(f',t)) \tag{3}$$

$$\forall f,t.term(f,t) \equiv \exists a,g.happens(a,t) \wedge terminates(a,f,g) \wedge (\forall f'.g(f') \rightarrow holds(f',t)) \tag{4}$$

We now write the three conditions about termination/initiation points.

$$\forall f,t_1,t_3.(\forall t_2.t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow \neg(term(f,t_2) \vee init(f,t_2))) \rightarrow \quad holds(f,t_1) \equiv holds(f,t_3) \tag{5}$$

$$\forall t_1,t_3,f.init(f,t_1) \wedge (\forall t_2.t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow \neg term(f,t_2)) \rightarrow holds(f,t_3) \tag{6}$$

[1] Note that we do not have the intuitive

$$\forall f,t,g.initiates(a,f,g) \equiv terminates(a,not(f),g),$$

as this would change the requirements on domain descriptions being consistent. That is, the domain description,

A **terminates** F_1 **when** F A **initiates** F_1 **when** F

has models.

$$\forall t_1, t_3, f.term(f, t_1) \wedge (\forall t_2.t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow \neg init(f, t_2)) \rightarrow \neg holds(f, t_3) \quad (7)$$

We also have unique names axioms for all our constants. We have domain closure axioms stating that every object is a constant, save for fluents, where we have an axiom that states that every fluent is a constant or the result of applying *not* to a fluent constant. We have axioms relating or negating the relation of every pair of time-points in the partial order $\leq$.

We now need to define the translation of a domain description in $\mathcal{E}$ into our second order language. To do this we need some definitions from Kakas and Miller, namely definitions 1, 2, 3, 4, 5, and 12.

We define the translation of a domain language [2] $\langle \Pi, \preceq, \Delta, \Phi \rangle$ into a set of sentences in second order logic. Our time-point constants are the objects in Π, our relation $t \leq t'$ is true when $t \preceq t'$. Our fluent constants are Φ, union the image of Φ under *not*. Our action constants are the set Δ. Our axioms are the unique names and domain closure axioms, the axiom 1, and the sentences defining $\leq$ for every pair of time-points. We denote the translation of a domain language $\mathcal{D}$ as $\mathcal{D}^*$.

Lemma: 1 *The theory of the sub-language of the translation of a domain language $\langle \Pi, \preceq, \Delta, \Phi \rangle$, whose only predicates are the equalities for each sort, and $\leq$, and whose functions are not and whose constants are all constants, is complete.*

Proof: Domain closure axioms give us quantifier elimination, all ground terms are decided by unique names axioms, save for fluents of the form $not(f)$, and $\leq$. Ground cases of $\leq$ are decided by assumption. This leaves equalities involving $not(f)$, which can be reduced to equalities of the form $f = not(f')$ for constant f, f', using $not(not(f)) = f$, and $not(f) = not(f') \rightarrow f = f'$. These cases are decided by 1. ∎

We define the translation of an h-proposition A **happens-at** T, as $happens(A, T)$. This is a well formed second order logic formula, in our language, as all time-points and actions in the domain language are constants in our second order language.

We define the translation of a t-proposition L **holds-at** T as $holds(L, T)$ if L is a constant, and as $\neg holds(F, T)$ if $L = \neg F$.

We define the translation of a c-proposition A **initiates** F **when** C, as

$$initiates\left(A, F, \lambda f. \bigvee_{F' \in C} f = F'\right).$$

We define the translation of a c-proposition A **terminates** F **when** C, as

$$terminates\left(A, F, \lambda f. \bigvee_{F' \in C} f = F'\right).$$

Finally we define the translation of an r-proposition L **whenever** C as,

$$whenever\left(F, \lambda f. \bigvee_{F' \in C} f = F'\right)$$

if L is a fluent constant F, and as $whenever\left(not(F), \lambda f. \bigvee_{F \in C} f = F\right)$ if L is $\neg F$ for some fluent constant F.

We define the translation of a domain description D, as the set of translations of its elements, and we denote it D^*. Given a set of propositions P, we define their translation P^*, by applying the above translation.

We use the usual notation $Circ(A; P)$ for the minimization of P in the theory A.

Lemma: 2 *A domain description, D, stated in $\mathcal{D}$, contains a h-proposition, A* **happens-at** *T if and only if $Circ(P^*; happens) \models happens(A, T)$, where P is the set of h-propositions in D.*

[2] Note that if Δ, Φ or Π are infinite, then we may not be able to write down domain closure axioms. Therefore we limit ourselves to the case where they are all finite. We can easily move to non-finite domains by introducing well-orders of each domain, which would allow us to state domain closure using induction on the appropriate well-order.

Proof:

The circumscription is equivalent to,

$$\forall a, t. happens(a, t) \equiv \bigvee_{A \textbf{ happens-at } T \in P} a = A \wedge t = T$$

The lemma follows immediately. ∎.

Lemma: 3 *A domain description, D, stated in $\mathcal{D}$ contains a c-proposition, A* **initiates** *F* **when** *C if and only if* $Circ(P^*; initiates) \models initiates\left(A, F, \lambda f. \bigvee_{F \in C} f = F\right)$*, where P is the set of c-propositions in D.*

Proof:

The circumscription is equivalent to,

$$\forall a, f, g. initiates(a, f, g) \equiv \bigvee_{A \textbf{ initiates } F \textbf{ when } C \in P} a = A \wedge f = F \wedge \forall f. g(f) \equiv \bigvee_{F \in C} f = F$$

The lemma follows immediately. ∎.

The analogous lemma also holds for *terminates*.

This gives the notion of a model of $\mathcal{E}$ without ramifications. We use Germanic font for second order structures, and use $holds^{\mathfrak{M}}$ for the interpretation of a predicate *holds* in the structure $\mathfrak{M}$.

Theorem: 1 *Let D be a domain description, stated in $\mathcal{D}$. Then its models are isomorphic to the interpretation of holds in models of $\mathcal{D}^*$, 1, 3 , 4, 5, 6, 7, D^*, $Circ(P^*; happens)$, $Circ(Q^*; initiates)$, $Circ(Q^*; terminates)$, where P and Q are the h, c propositions in D respectively.*

Proof: We consider the conditions on being a model of D, and show that they agree with the second order conditions.

Firstly, by lemma 1 the sorts, under the interpretation of $=$ and $\leq$, give a domain language, as by completeness there is a unique model that is exactly three sets and a partial ordering.

Secondly, *holds* is of the correct type to be an interpretation, namely a function from Φ, Δ to $\{$ true, false$\}$. Thus, we need only check the four conditions on being a model. However, these depend on the notion of an initiation/termination point. We first show that $init(F, T)$ ($term(F, T)$) is true in a model $\mathfrak{M}$, exactly then T is an initiation(termination) point for F relative to the interpretation $holds^{\mathfrak{M}}$.

We only consider initiation points as the argument for termination points is exactly analogous.

The definition of an initiation point is in terms of whether there is an h-proposition of the form A **happens-at** T, and a c-proposition of the form A **initiates** F **when** C. However, by Lemma 2 and Lemma 3, this is exactly equivalent to the truth of whether our circumscription implies $happens(A, T)$, and $initiates\left(A, F, \lambda f. \bigvee_{F \in C} f = F\right)$. There are no other sentences that mention *happens* or *initiates* save for the circumscription, and D^*. However, if D^* is consistent, then D^* implies $happens(A, T)$ only if the circumcription implies it. Therefore, if the sentences, $\mathcal{D}^*$, 1, 3, 4, 5, 6, 7 are consistent, then the entire theory entails, $happens(A, T)$ if and only if there is an h-proposition of the form A **happens-at** T. The sentences $\mathcal{D}^*$, 1 are immediately consistent, as the domain language provides a model. 4 and 3 can be taken to be definitions, and are thus their addition does not make the other sentences $\mathcal{D}^*$ and 1 inconsistent. Thus we need to show that 5 6 and 7 are consistent with $\mathcal{D}^*$, 1 and 4 and 3. If D has an interpretation, this serves as a model. Thus we may take the sentences to be consistent. Thus we have shown that the entire theory entails, $happens(A, T)$ if and only if there is an h-proposition of the form A **happens-at** T. The case of *initiates* is similar.

Thus, we are reduced to checking that the interpretation satisfies C at T. However, this is equivalent to $\forall f. \bigvee_{F \in C} f = F \rightarrow holds(f, T)$, and thus the existence of an initiation point is equivalent to whether $\exists a, g. happens(a, T) \wedge initiates(a, F, g) \wedge (\forall f'. g(f') \rightarrow holds(f', T))$ is true. This is the definition of $init(F, T)$, and thus $init(F, T)$ is true in $\mathfrak{M}$ if and only if T is an initiation point for F in $holds^{\mathfrak{M}}$.

Now we can check our four conditions. The first is immediately 5, as $T \preceq T'$ when $T \leq T'$, *holds* is defined to be equal to the interpretation, and by the above *init* and *term* are true when there are initiation and termination points. The second and third are exactly 6 and 7, by the same reasoning.

Thus we are left with the fourth condition, which follows as the interpretation is defined in terms of $holds$.

Thus, the conditions for an interpretation H to be a model of a domain description D stated in $\mathcal{D}$ are exactly the conditions for a set of fluent, time-points pairs to be the interpretation of $holds$, in models that satisfy $\mathcal{D}^*$, 1, 3, 4, 5, 6, 7, $Circ(P^*; happens)$, $Circ(Q^*; initiates)$, $Circ(Q^*; terminates)$, D^* where P and Q are the h, c propositions in D respectively. ∎

To get ramifications we need to add the predicate $whenever$. We also add an axiom,

$$\forall f, g, t.whenever(f, g) \rightarrow ((\forall f'.g(f') \rightarrow holds(f', t)) \rightarrow holds(f, t).$$

We also need to change the definitions of $term$ and $init$.

We have a choice here, we can use the following definition, where $term$ is defined as any predicate satisfying,

$$\forall f, t.term(f, t) \equiv \begin{array}{c} (\exists a, g.terminates(a, f, g) \wedge \forall f'.g(f') \rightarrow holds(f', t)) \vee \\ \exists C_1, C_2.whenever(not(f), \lambda f'.C_1(f') \vee C_2(f')) \wedge \\ (\forall f'.C_1(f') \rightarrow holds(f', t)) \wedge \quad \forall f'.C_2(f') \rightarrow term(f', t) \end{array}$$

and $init$ is defined as any predicate satisfying,

$$\forall f, t.init(f, t) \equiv \begin{array}{c} (\exists a, g.initiates(a, f, g) \wedge \forall f'.g(f') \rightarrow holds(f', t)) \vee \\ \exists C_1, C_2.whenever(not(f), \lambda f'.C_1(f') \vee C_2(f')) \wedge \\ (\forall f'.C_1(f') \rightarrow holds(f', t)) \wedge \quad \forall f'.C_2(f') \rightarrow init(f', t). \end{array}$$

This suffers from the counter-example suggested by Thielscher. However, my reading of the paper suggests that we take the smallest relations $term$ and $init$ that satisfy the above, as the definition, as the authors suggest that the least fixed-point be taken.

The proof that this captures the same notions as the definition of a model in $\mathcal{E}$ is essentially the same as before, save that we have some more conditions to check.