A formal analysis of KGP agents

F. Sadri and F. Toni

Department of Computing, Imperial College London,
{fs,ft}@doc.ic.ac.uk

Abstract. This paper contributes to the identification, formalisation and analysis of desirable properties of agent models in general and of the *KGP* model in particular. This model is specified in computational logic, and consequently lends itself well to formal analysis. We formalise three notions of welfare, in terms of goal achievement, progress, and reactive awareness, and we prove results related to these notions for KGP agents. These results broadly demonstrate the coherence of some of the design decisions in the KGP model, the need for some of its components for effectiveness in goal achievement, the extent to which the welfare of KGP agents can be shown to improve during their life-time, and the awareness of the agents of their reactions to changes in the environment.

1 Introduction

The use of logic to formalise and prove formal properties of agent models has been advocated by several researchers in the filed of agents. The KGP model of agency [9, 7] was designed with these aims in mind, as well as allowing agents with proactive and reactive behaviour in a dynamic environment. The KGP model is modular and allows for design of heterogeneous agents, each equipped with its own profile. Agents in the KGP model are equipped with knowledge bases, capabilities and transition rules that allow them to plan for their goals, make observations in the environment in which they are situated, update their beliefs, react to changes in their environment, communicate with other KGP agents, revise their states, and dynamically change their goals. The model has been described in detail in [9, 7] and compared with other models of agency, for example IMPACT [1], BDI [16], 3APL [6], AgentSpeak [15] and MINERVA [11]. All the components of the KGP model have been specified using computational logic. This was done to facilitate formalisation and verification of formal properties in addition to enabling a verifiable implementation of the model [4, 3]. In this paper we focus on the former, and propose three notions of agent welfare: *goal achievement*, referring to the achievement of goals held by an agent; *progress*, referring to how close an agent may be to achieving its goals; *reactive awareness*, referring to how aware an agent is of reactions that are necessary to its circumstances and environment.

Our notions of welfare amount to assessing how effective an agent is in achieving its goals, or at least in working towards achieving them, and in reacting to its decisions and environment. This work, therefore, is significant in allowing us to analyse the effectiveness of the KGP model. Also, a study of the environmental or internal conditions that would help, guarantee or hinder improvement of welfare, could help give guidelines to designers of agents.

In our analysis we mostly adopt a *subjective* approach to the notion of welfare, whereby, for example, achievement of a goal is assessed wrt the agent's beliefs (knowledge base). We briefly discuss, in sections 4 and 5, *objective* notions of welfare, wrt the agent's actual environment, rather than its perception of it.

The paper is structured as follows. In Section 2 we give the abstract agent model and its instantiation as the KGP model. In Section 3 we give preliminary definitions and results, used in the rest of the paper. In Sections 4, 5 and 6 we study the concepts of goal achievement, progress and reactive awareness. In Section 7 we conclude with some related work.

2 Agent model

We will assume an agent model, generalising the KGP model, whereby agents are equipped with the following components:

- an *internal (or mental) state*, consisting of the agent's *beliefs*, *goals* and *plans*; goals and plan components have associated times and temporal constraints, inducing a partial order;
- a set of *reasoning capabilities*, reasoning with the information available in the agent state;
- a *Sensing capability*, allowing agents to observe their environment and actions by other agents;
- a set of *transition rules*, changing the agent's state; the transition rules are defined in terms of the capabilities, and their effect is dependent on the concrete time of their application;
- a set of *selection functions* to select inputs to transitions from the state;
- a *control*, for deciding which enabled transition should be next, based on the selection functions, the current time, and the previous transition.

The application of a transition T at time τ, mapping state S onto S' given (a possibly empty) input X, will be represented as $T(S, X, S', \tau)$. We will assume for simplicity that the application of transitions is instantaneous, namely τ is also the time when S' is generated.

The control of the agent is responsible for its behaviour, in that it induces an *operational trace*, namely a (typically infinite) sequence of transitions

$$T_1(S_0, X_1, S_1, \tau_1), \ldots, T_i(S_{i-1}, X_i, S_i, \tau_i),\ T_{i+1}(S_i, X_{i+1}, S_{i+1}, \tau_{i+1}), \ldots$$

such that S_0 is the given initial state, and for each $i \geq 1$, τ_i is given by the clock of the system and $\tau_i < \tau_{i+i}$, (namely time increases).

State in KGP agents: $\langle KB, Goals, Plan, TCS \rangle$. The KB component holds the agent's *beliefs*. It includes a dynamic part KB_0, updated when the agent observes the environment (via its Sensing capability) and executes actions in plans. In particular, the fact that action $a[t]$ has been executed at time τ is recorded by means of $executed(a[t], \tau)$. KB also includes knowledge bases to support the various reasoning capabilities, as we will discuss later.

The $Plan$ consists of *plan-trees* whose roots are goals in $Goals$ or simply actions [1], and whose non-root nodes are actions or sub-goals. In a plan-tree, the set of all children of a node form a (partial) plan for the node, the set of leaves of any sub-tree form a (partial) plan for the root of the sub-tree, and actions are leaves. Each goal in $Goals$ is the root of at most one plan-tree.

The TCS component is a set of (temporal) constraints, built from operators including $<, \leq, =, \neq$. Goals, actions and sub-goals have time parameters constrained by TCS. These temporal constraints induce a partial order on actions and sub-goals in plans.

(Sub-)Goals are *timed fluent literals* of the form $l[t]$, where l is a fluent literal (property) of the form p or $\neg p$ and t is its associated time, and actions are *timed action literals* of the form $a[t]$, where a is an action operator and t is its associated time. Implicitly, all time variables are existentially quantified within the agent's state. Actions may be "physical", communicative or sensing, and fluents may be "mental" (to be brought about by plans) or sensing (to be observed).

Reasoning capabilities in KGP agents. In KGP agents, the main reasoning capabilities support Planning, Temporal Reasoning, Reactivity, Goal Decision, and Temporal Constraint Satisfiability. The Planning, Reactivity and Goal Decision capabilities all need to incorporate Temporal Constraint Satisfiability within them. This is to ensure that the agent state is always "consistent" as it is updated by means of goals, sub-goals and actions generated by these capabilities.

We will indicate with $\models_{tcs} C$ that the set of temporal constraints C is satisfiable by means of the Temporal Constraint Satisfiability capability. Intuitively, satisfiability amounts to the existence of a concrete instantiation of the variables in C that render C true wrt the underlying domain for the evaluation of the constraint operators. Thus, e.g. for the integers, $\models_{tcs} 3 \leq t$ but $\not\models_{tcs} 3 \leq t \wedge t < 2$.

Given a state $S = \langle KB, Goals, Plan, TCS \rangle$ and a concrete time point τ, we will use the notation $S, Y \models^{\tau}_{cap} Z$ to indicate, intuitively, that Z is "generated" as the result of the application of capability cap at time τ in state S, where cap is one amongst $plan$ (for Planning), tr (for Temporal Reasoning), $react$ (for Reactivity) and gd (for Goal Decision). Formally, $S, Y \models^{\tau}_{cap} Z$ stands for $KB_{cap} \cup KB_0 \cup X, Y \models^{\tau}_{cap} Z$, where:

- for $cap = plan$, X is $Plan$, Y is a set of (sub-)goals in $Plan$ to be planned for, together with TCS, and Z is either a plan for Y (consisting of a set of actions/sub-goals and a new set of temporal constraints), or $\perp$, representing that no such plan exists;
- for $cap = tr$, X and Y are empty and Z is a timed fluent literal together with some temporal constraints including TCS; intuitively, this capability is used to verify whether or not the literal in Z holds, at a time point satisfying the temporal constraints in Z;
- for $cap = react$, X is $Plan$ and TCS, Y is empty, and Z is either a set of actions/goals and a new set of temporal constraints or $\perp$, representing that no such reaction exists;

[1] If an action is the root of a plan-tree, that action is necessarily an outcome of reacting, using the knowledge base that supports the Reactivity capability.

– for $cap = gd$, X and Y are empty and Z is a set of timed fluent literals (representing new goals) and a new set of temporal constraints.

In the sequel, when Y is empty we will simply drop it.
The outcome of the capabilities is affected by the current time τ, e.g., in the case of Planning and Reactivity, because the generated actions need to be executable in the future (after τ), or, in the case of Goal Decision, because the generated goals need to be achievable in the future.
For $cap = plan$, if Z consists solely of actions, Z is called a *full plan* for Y.
In every state the KB of the agent, in addition to KB_0, includes a modular collection of specialised knowledge bases. These are KB_{plan}, KB_{gd}, KB_{tr}, KB_{react}, used, respectively, by the Planning, Goal Decision, Temporal Reasoning and Reactivity capabilities. KB_{plan}, for example, may contain a plan library or a theory of action and causality such as the event calculus [10]. Independently of the concrete realisation choices for these modules and the corresponding capabilities, we will assume that KB_{plan} and KB_{tr} in KB in any $S = \langle KB, Goals, Plan, TCS \rangle$ are related in such a way that, informally, if $S, g[t] \models^{\tau}_{plan} P$, for $P \neq \bot$ and a full plan, then $S' \models^{\tau'}_{tr} g[t]$, where S' is the state S after having executed all actions in P at times satisfying TCS and all constraints in P and τ' is after τ.

Transition rules in KGP agents. Transitions affect the agent's state and are defined in terms of the capabilities, as follows:

– *Goal Introduction (GI)*, changing the $Goals$ and TCS, using Goal Decision, and changing $Plan$, by adding one plan-tree (consisting solely of the root) for each new goal in $Goals$;
– *Plan Introduction (PI)*, changing the $Plan$, by adding children to (sub-)goals, and changing TCS, and using Planning;
– *Reactivity (RE)*, changing $Goals$, by adding any new "reactive goals", $Plan$, by adding one plan-tree (consisting solely of the root) for each new goal in $Goals$ and any new "reactive actions", and TCS; the new goals, actions and temporal constraints are obtained by using Reactivity;
– *Sensing Introduction (SI)*, changing $Plan$ and TCS, by introducing new (temporally constrained) *sensing actions*, e.g. for checking the preconditions of actions already in $Plan$, and using Sensing;
– *Passive Observation Introduction (POI)*, changing KB_0 by introducing information coming from the environment without being actively sought by the agent, and using Sensing;
– *Active Observation Introduction (AOI)*, changing KB_0 by introducing actively sought information from the environment, and using Sensing; this information might be needed, for example, to confirm that actions have been successfully executed;
– *Action Execution (AE)*, executing actions, and changing KB_0;
– *State Revision (SR)*, revising $Plan$, and using Temporal Reasoning and Temporal Constraint Satisfiability. In particular, SR deletes from $Plan$ all achieved or timed-out (sub-)goals, as well as all their descendents in the plan-trees in $Plan$, and all executed or timed-out actions. It also deletes all descendents of (sub-)goals with one or more timed-out children, thus eliminating plans which have no chance of success.

The effect of transitions is dependent on the concrete time of their application, taken into account by the capabilities called therein.

Selection functions in KGP agents. These include c_{GS} and c_{AS}, to select goals and sub-goals to be planned for and actions to be executed, respectively, and c_{FS} and c_{PS}, to select fluents to be sensed immediately, by AOI, and fluents to be sensed eventually, by SI, respectively. These functions provide appropriate inputs to (some of) the transitions and enable them, as discussed below.

Control in KGP agents. The operational traces are not fixed a priori, as in conventional agent architectures, but are determined dynamically by reasoning with declarative cycle theories, giving a form of flexible control. In this paper, we do not provide details of these cycle theories (see [9, 7, 8, 18]). Here, it suffices to say that the cycle theory induces an operational trace

$T_1(S_0, X_1, S_1, \tau_1), \ldots, T_i(S_{i-1}, X_i, S_i, \tau_i), T_{i+1}(S_i, X_{i+1}, S_{i+1}, \tau_{i+1}), \ldots$

such that T_i is one of the KGP transitions, and X_i is an input as follows

- if T_i is one of GI, RE, POI, SR, then X_i is empty;
- if T_i is PI, then X_i is non-empty and it is the set of all (sub-)goals determined by c_{GS}; these are (sub-)goals to be planned for;
- if T_i is AE, then X_i is non-empty and it is the set of all actions determined by c_{AS}; these are actions to be executed;
- if T_i is SI, then X_i is non-empty and it is the set of all fluents determined by c_{PS}; these are fluents to be actively sensed in the future, e.g. preconditions of actions;
- if T_i is AOI, the X_i is non-empty and it is the set of all fluents determined by c_{FS}; these are fluents to be actively sensed immediately.

We say that the control is *fair* iff no transition is ever postponed indefinitely.

3 Preliminaries

In this section we show how certain design choices for KGP agents allow to prove some basic, desirable properties of KGP agents needed to prove later results.

Definition 1. *Given a state $S = \langle KB, Goals, Plan, TCS \rangle$ and a time point τ:*

- *a goal, action or sub-goal $z[t]$ is* timed-out *at τ iff $\not\models_{tcs} t \geq \tau \wedge TCS$;*
- *an action or sub-goal $z[t]$* belongs to a plan *for a goal or sub-goal $g[t']$ iff $z[t]$ is a descendent of $g[t']$ in a plan-tree in Plan;*
- *a goal or sub-goal $g[t]$ is* believed to be achieved *in S at τ iff $S \models^{\tau}_{tr} g[t] \wedge TCS \wedge t \leq \tau$;*
- *two actions $a_1[t_1]$ and $a_2[t_2]$ in any plan-trees in Plan are said to be* incompatible *at τ iff $\not\models_{tcs} TCS \wedge t_1 = \tau \wedge t_2 = \tau$;*
- *a timed (fluent literal or action) literal $x[t]$* matches *a timed (fluent literal or action) literal $x[t']$ in S iff $\models_{tcs} t = t' \wedge TCS$;*
- *a timed action literal $a[t]$* is executed in *S iff $executed(a[t'], \tau) \in KB_0$ and $a[t]$ matches $a[t']$ in S.*

The following is a property of KGP agents that they do not attempt to execute actions that they believe are infeasible or unnecessary, and do not attempt to plan for goals if a plan is not needed or if it is too late to plan for them.

Theorem 1.

- *KGP agents never attempt to execute actions that*
  - *are timed-out, or*
  - *have an ancestor or the child of an ancestor that is timed-out, or*
  - *belong to a sub-tree for a goal that they believe is achieved, or*
  - *have an ancestor that they believe is achieved, or*
  - *have a precondition whose complement they believe is achieved;*
- *KGP agents never attempt to plan for a goal or a sub-goal that*
  - *already has children in a plan-tree, or*
  - *is timed-out or that they believe is already achieved, or*
  - *belongs to a plan-tree for a goal that is timed-out or that they believe is achieved, or*
  - *has an ancestor that is timed-out or that they believe is achieved.*

This result follows from the definition of action selection and goal selection functions. The following result is another consequence of the definition of action selection function.

Theorem 2. *Incompatible actions are never executed concurrently.*

4 Goal achievement

In this and section 5, to simplify the presentation, we ignore the RE transition. This section builds upon, in part, some of the work reported in [14].
Given a (possibly infinite) operational trace for an agent:
$T_1(S_0, X_1, S_1, \tau_1), \ldots, T_j(S_{j-1}, X_j, S_j, \tau_j), \ldots, T_l(S_{l-1}, X_l, S_l, \tau_l), \ldots$
with $0 \leq j < l$, we refer to the (possibly infinite) sequence of states:
$S_0, S_1, \ldots, S_{j-1}, S_j, \ldots, S_{l-1}, S_l, \ldots$
as the *state-sequence* (of the trace), and to any (possibly infinite) sub-sequence
$S_{j-1}, S_j, \ldots, S_{l-1}, S_l, \ldots$
of a state-sequence as a *portion* (of the state-sequence). We also refer to the time τ_j at which a state S_j is generated in a trace $\ldots, T_j(S_{j-1}, X_j, S_j, \tau_j), \ldots$ giving a state sequence $SS = \ldots S_{j-1}, S_j \ldots$ as $time(S_j, SS)$.
The following definition gives the criterion according to which we judge a state-sequence or portion as providing successive improvements over states. It is parametric wrt a notion of preference $\ll$ between states. Note that this definition is somewhat naive, as will be explained later.

Definition 2. *Let $\ll$ be any notion of preference between states.*

- *An* infinite *state-sequence or portion* $S_0, S_1, \ldots, S_n, \ldots$ improves welfare wrt $\ll$ *iff for each $j \geq 0$, there exists $l > j$ such that $S_j \ll S_l$.*
- *A* finite *state-sequence or portion* $S_0, S_1, \ldots, S_n$ improves welfare wrt $\ll$ *iff for each $j \geq 0$, $j < n$, there exists $l > j$, $l \leq n$ such that $S_j \ll S_l$.*
- *An agent is* $\ll$-improving wrt some initial state *iff the state-sequence corresponding to any operational trace induced by its control, from the given initial state, improves welfare wrt $\ll$.*

Note that this definition does not impose any condition on intermediate states between S_j and S_l, and in particular any such state might actually bring the agent into a worse state than S_j, wrt $\ll$. Stronger notions could be adopted, for example that for each $j \geq 0$, for each $l > j$, $S_j \ll S_l$. However, we believe that such stronger notions would be too limiting in practice. Note also that we could define a much weaker notion of $\ll$-improvement for an agent, requiring only for it to be $\ll$-improving wrt *some* given class of initial states.
For every concrete notion of preference between states we obtain a concrete notion of improvement in definition 2. One such notion of preference, that we will refer to as $\ll_1$, sanctions, informally, that $S \ll_1 S'$ iff in S' at least as many goals have been achieved as in S. Another such notion, that we will refer to as $\ll_2$, sanctions that $S \ll_2 S'$ iff in S' strictly more goals have been achieved than in S. There are clear connections between these notions of achievement and modelling the preferences of agents using utility functions, as the number of achieved goals gives a very simple kind of utility function. Formally:

Definition 3. *Given a state* $S = \langle KB, Goals, Plan, TCS \rangle$ *and a time* τ, *we define the* number of achieved goals in S at τ as
$A^+(S,\tau) = \#\{l[t] \mid l[t] \in Goals \text{ and } l[t] \text{ is believed to be achieved in } S \text{ at } \tau\}$.
Then, given a (portion of a) state-sequence SS *and states* S, S' *in* SS *with* $\tau = time(S, SS)$ *and* $\tau' = time(S', SS)$*:*
- $S \ll_1 S'$ *iff* $A^+(S,\tau) \leq A^+(S',\tau')$*;*
- $S \ll_2 S'$ *iff* $A^+(S,\tau) < A^+(S',\tau')$.

Intuitively, (the designer of) an agent adopting $\ll_1$ ($\ll_2$) believes that its welfare can be improved by never decreasing (always increasing) the number of achieved goals. Note that we take a subjective view of achievement: goals are achieved if they can be proven subjectively by the agent via its Temporal Reasoning capability. There are alternative notions of achievement that we could have adopted, e.g. a stronger subjective notion, whereby only immediately after SR the agent can evaluate its achievement, or a fully objective notion, whereby it is some "external observer", knowing exactly what holds and does not hold in the environment, who evaluates whether goals are achieved and when via its own "temporal reasoning capability" wrt its complete knowledge of the environment. Finally, note that other choices of $\ll$ would have been possible, e.g. by considering the number of unachievable goals or *ratio* between achieved and unachievable goals.

Example 1. Assume g_1, g_2, g_3 are timed fluent literals. The following is a possible (finite portion of a) state-sequence starting with S_0 with $KB_0 = Goals = Plan = \{\}$, with the associated values of A^+ (but ignoring the time of states). Here, we indicate with "_" components that we ignore for simplicity as irrelevant.

$S_0 = \langle \{\}, \{\}, \{\}, \{\} \rangle$ $\qquad A^+(S_0) = 0$
T_1 is GI:
$S_1 = \langle \{\}, \{g_1, g_2, g_3\}, \{g_1, g_2, g_3\}, \_ \rangle$ $\qquad A^+(S_1) = 0$
T_2 is POI, leading to g_1 holding:

$S_2 = \langle \_, \{g_1, g_2, g_3\}, \{g_1, g_2, g_3\}, \_\rangle$ $A^+(S_2) = 1$
T_3 is PI for g_3, introducing a full plan:
$S_3 = \langle \_, \{g_1, g_2, g_3\}, \_, \_\rangle$ $A^+(S_3) = 1$
T_4 is AE, executing all actions for g_3 in *Plan*:
$S_4 = \langle \{\}, \{g_1, g_2, g_3\}, \_, \_\rangle$ $A^+(S_4) = 2$
T_5 is SR (g_1, g_3 achieved and g_2 timed-out):
$S_5 = \langle \_, \{g_1, g_2, g_3\}, \{\}, \_\rangle$ $A^+(S_5) = 2$

Here, every state is better that any earlier state wrt $\ll_1$, thus this trace is $\ll_1$-improving. However, it is not $\ll_2$-improving (e.g. S_4 cannot be improved upon).

So far we have assumed that any two states in a state-sequence or portion of it can be compared using $\ll$. This is inappropriate when the GI transition modifies the *Goals* in a state and after all (sub-)goals have been achieved or become timed-out, as illustrated by the next example wrt the concrete notion of $\ll_2$.

Example 2. Assume the following state-sequence (with associated A^+):

$S_0 = \langle \_, \{g_1\}, \_, \_\rangle$ $A^+(S_0) = 0$ (g_1 not achieved yet)
$S_1 = \langle \_, \{g_1\}, \_, \_\rangle$ $A^+(S_1) = 0$ (g_1 not achieved yet)
$S_2 = \langle \_, \{g_3, g_4\}, \_, \_\rangle$ $A^+(S_2) = 0$ (g_1 dropped, g_3, g_4 introduced by GI and not achieved yet)
$S_3 = \langle \_, \{g_3, g_4\}, \_, \_\rangle$ $A^+(S_3) = 1$ (g_1 achieved)

Here, g_1 may be (believed to be) achieved because of a POI. According to definition 2, $S_0, \ldots, S_3$ is $\ll_2$-improving, which is counter-intuitive, since the agent should not be better off at achieving goals that it has dropped in favour of newly adopted goals. Thus, $A^+(S_3)$ should be the number of goals in $\{g_3, g_4\}$ that are believed to be achieved. Consider now the state-sequence S_0, S_1, S_2 followed by

$S'_3 = \langle \_, \{g_3, g_4\}, \_, \_\rangle$ $A^+(S'_3) = 1$ (g_3 achieved)
$S_4 = \langle \_, \{g_3, g_4\}, \_, \_\rangle$ $A^+(S_4) = 1$ (g_3 achieved, g_4 timed-out)
$S_5 = \langle \_, \{g_3, g_4\}, \_, \_\rangle$ $A^+(S_4) = 1$ (g_3 achieved, g_4 timed-out)
$S_6 = \langle \_, \{g_3, g_4\}, \_, \_\rangle$ $A^+(S_4) = 1$ (g_3, g_4 dropped by SR)

S_5 might be the outcome of a POI. According to definition 2, $S_0, \ldots, S_6$ is not $\ll_2$-improving, which is counter-intuitive, since the agent has done its best to achieve as many goals as possible and reach state S_4. This is the last state that should "count" as far as $\ll$ is concerned.

The notion of $\ll$-improvement can be easily modified to look at portions related to the same *Goals* and ignoring states following other states with all goals in *Goals* either achieved or timed-out. We omit this definition here for lack of space.

Theorem 3. *Any KGP agent is $\ll_1$-improving.*

This result holds because of the features of the KGP model, according to which goals, once achieved, can never become "unachieved". This is due to the fact that goals are existentially quantified in the model, and that the agents do not observe information about the past. If at τ an agent believes that a goal holds

at some time t, then there is an instance τ' of t that satisfies the temporal constraints, τ' is before τ and the agent believes the goal held at τ'. Then, in all future states the agent will continue to believe that the goal held at τ'.
The analogous result for $\ll_2$ does not hold, e.g. see example 1. However, we can prove the following results regarding $\ll_2$, given a state sequence SS and states $S = \langle KB, Goals, Plan, TCS \rangle$ and $S' = \langle KB', Goals', Plan', TCS' \rangle$ in SS playing the role of (and satisfying the conditions for) S_j and S_l in definition 2.

Theorem 4. *If $S \ll_2 S'$, then $KB \subset KB'$ and, in particular, $KB_0 \subset KB'_0$.*

This theorem shows the importance of sensing the environment and executing actions to improve welfare according to $\ll_2$. This is made more explicit by theorem 6. Note that KB_0, holding the outcome of the agent's sensing of the environment and the recording of any action executed by the agent, is the only part of the agent KB that is dynamically modified.

Theorem 5. *If $S \ll_2 S'$, then there exists a state S'' such that either S'' is in between (but different from) S and S' in SS or $S'' = S'$ and there exists a (sub-)goal $g[t]$ in S'' such that $g[t]$ is believed to be achieved in S'', but not in S.*

As a consequence of theorem 5 we can show:

Theorem 6. *If $S \ll_2 S'$, there is a state S'' in between (but different from) S and S' in SS such that in S'' one of POI or AE or PI has been performed.*

This theorem shows the importance of the three transitions, POI, AE and PI, in improving welfare in terms of goal achievement. However, because of time criticality and the dynamism of the environment, these transitions cannot guarantee achievement of all goals. But they help *make progress* towards achievement of goals in a sense that will be formalised in section 5.
In general, we cannot guarantee that all achievable goals will eventually be achieved, namely that the maximum element of either of the $\ll_1$ and $\ll_2$ orderings can be found. There are three main reasons for this: (i) goal are temporally constrained and it may not be possible to achieve them by their deadlines, (ii) the environment can be unpredictable, (iii) the agent may not know of compatible plans for the goals given what it believes about the environment. Goals can be *guaranteed* to be achieved under some restrictive conditions, omitted here for lack of space.

5 Progress

In this section we define a new ordering between states, based on a notion of "progress" and relate this ordering to the one induced by A^+. We also show how some of the transitions in the KGP model affect progress. For simplicity, here we will assume that the Planning capability always produces full plans.

Definition 4. *A state $S = \langle KB, Goals, Plan, TCS \rangle$ at time τ* potentially achieves *a goal $g[t] \in Goals$ with a set of timed actions $A = \{a_1[t_1], \ldots, a_n[t_n]\}$ iff*

- *$g[t]$ is not already believed to be achieved in S at τ, and*
- *A is a set of actions in $Plan$, and*
- *no action in A has already been executed in S, and*
- *there exist concrete times $\tau', \tau_1, \ldots, \tau_n$ such that*
  - $\models_{tcs} \tau < \tau_1 \leq \tau' \wedge \ldots \wedge \tau < \tau_n \leq \tau'$*, and*
  - $\models_{tcs} t_1 = \tau_1 \wedge \ldots \wedge t_n = \tau_n$*, and*
  - $KB \cup \{executed(a_1[t_1], \tau_1), \ldots, executed(a_n[t_n], \tau_n)\} \models^{\tau}_{tr} g[\tau']$.

Definition 5. *Given states S and S' and times τ, τ' and a goal $g[t]$, we say that $S \prec_{g[t]} S'$ iff*

- *S potentially achieves $g[t]$ with some A and*
- *S' potentially achieves $g[t]$ with some A' and*
- *$A' \subset A$ and*
- *S does not potentially achieve $g[t]$ with A'.*

Intuitively, if $S \prec_{g[t]} S'$ then S' is a more progressed state than S as far as the achievement of $g[t]$ is concerned, since there are fewer actions still to be executed before $g[t]$ is actually achieved (if all goes according to the $Plan$).
In a state with maximal goal achievement, either all goals are achieved or no more progress is possible, namely:

Theorem 7. *Given a state sequence SS, for any S in SS if there is no state S' in SS such that $S \ll_2 S'$, then*
either all the goals in S are believed to be achieved at $time(S, SS)$
or for no goal $g[t]$ in S there exists a state S'' such that $S \prec_{g[t]} S''$.

Note that S'' does not need to be in SS.
The following theorem sanctions that AE improves progress towards achievement. It follows from theorems 1 and 2.

Theorem 8. *Let $T_1(S_0, X_1, S_1, \tau_1), \ldots, T_i(S_{i-1}, X_i, S_i, \tau_i), \ldots$ be any trace. If T_i is AE then*
either all the goals in S_i are believed to be achieved at τ_i
or there exists $g[t]$ such that $S_{i-1} \prec_{g[t]} S_i$.

The following theorem sanctions that PI is needed in order to pave the ground for progress:

Theorem 9. *Let SS be a state sequence and S, S' be two states in SS such that $time(S, SS) < time(S', SS)$. Then, if there exists $g[t]$ such that $S \prec_{g[t]} S'$, then there exists S'' resulting from a PI such that $time(S'', SS) < time(S', SS)$.*

6 Reactive awareness

Reactivity is a major feature of the KGP model. It allows condition-action type of behaviour and some element of dynamic plan repair. It also allows to generate dynamically obligations and prohibitions [17]. Next we define R^+, which intuitively gives a measure of how "aware" the agent is of its reactive necessities.

Definition 6. *Given a state* $S = \langle KB, Goals, Plan, TCS\rangle$ *and a time* τ*, let*

- $Set_1(S,\tau) = \{x[t] | S \models^{\tau}_{react} x[t]\}$
- $Set_2(S,\tau) = \{x[t] | x[t] \in Set_1(S,\tau)$ *and*
  $x[t]$ *matches a (fluent or action) literal in* $Plan$ *or*
  $x[t]$ *is believed to be achieved in* S *at* τ *or*
  $x[t]$ *is executed in* $S\}$*.*

Then, $R^+(S,\tau) = \dfrac{\#Set_2(S,\tau)}{\#Set_1(S,\tau)}$.

Intuitively, $Set_1(S,\tau)$ represents all the actions that have to be executed and all the goals that have to be achieved in reaction to the circumstances the agent finds itself in state S at time τ (according to its KB_{react}). $Set_2(S,\tau)$ represents the subset of $Set_1(S,\tau)$ that the agent is "explicitly aware" of, namely "reactive actions" that it has already executed or at least included in its $Plan$ to execute, and "reactive goals" that it has already achieved or included in its $Plan$ to achieve. Then the ratio R^+ gives a measure of reactive awareness.
The Reactivity capability in the KGP model is designed so that, if it is possible to have mutually consistent reactions, it produces all the necessary reactions (i.e. R^+=1). This is captured by the following theorem.

Theorem 10. *Let* $T_1(S_0, X_1, S_1, \tau_1), \ldots, T_i(S_{i-1}, X_i, S_i, \tau_i), \ldots$ *be any trace. If* T_i *is RE,* $S_{i-1} \models^{\tau_i}_{react} Z$ *and* $Z \neq \bot$*, then* $R^+(S_i, \tau_i) = 1$*.*

Alternative design choices might be suitable, in the case of an inconsistent set of reactions, to guarantee as high a value of R^+ as possible, e.g. by returning a maximally consistent subset of the set of inconsistent reactions, or by imposing preferences over reactive rules to eliminate inconsistencies.
The result in theorem 10 cannot be guaranteed after all the transitions, and in particular, after POI, AOI, GI, PI and SI. Indeed, POI and AOI might introduce new observations into the (KB_0 part of the) state, PI and SI will typically produce new actions and sub-goals and GI might introduce new goals in the state, and the Reactivity capability might require that some new reactions are introduced to take into account these extensions to the state. However, it is natural in general that R^+ fluctuates, because it depends heavily on the dynamics of the environment and the agent.
An agent equipped with a fair control will always reach, at some point, a state wrt maximal R^+, if one such state exists (i.e. in the absence of inconsistencies).

Theorem 11. *If the agent control is fair, then, for every operational trace, for every state* S *in the trace, there exists a later state* S' *in the trace such that*
either $R^+(S', time(S'))$ *is not defined*
or $R^+(S', time(S')) = 1$*.*

Note that an agent that is maximally achieved wrt A^+ can still work towards reaching an ideal R^+. Note also that the notion of R^+ above is *subjective*, in that it only considers reactions to what the agent believes about its environment (as well as its $Plan$ and $Goals$). We can also define an objective notion of R^+:

Definition 7. *Let E represent* complete information *about the environment. Then, given a time τ and a state $S = \langle KB, Goals, Plan, TCS\rangle$, let S_E be $\langle KB - KB_0 \cup E, Goals, Plan, TCS\rangle$ and let*

- $Set_1(S, E, \tau) = \{x[t] | S_E \models^{\tau}_{react} x[t]\}$
- $Set_2(S, E, \tau) = \{x[t] | x[t] \in Set_1(S, E, \tau)$ *and*
  $x[t]$ *matches a fluent or action literal in S or*
  $x[t]$ *is believed to be achieved in S at τ*
  $x[t]$ *is executed in S*$\}$.

Then, $R^+(S, E, \tau) = \dfrac{\#Set_2(S, E, \tau)}{\#Set_1(S, E, \tau)}$.

It is possible to design the control of the agent in order to guarantee maximal values of this new notion of R^+, mirroring theorems 10 and 11 above. This control needs to ensure that any RE transition is immediately preceded by an AOI transition, sensing fluents that are triggers in reactive rules in KB_{react}. In a reactive rule $l[t] \Rightarrow a[t'] \wedge t' < t + 10$ in KB_{react} $l[t]$ is one such trigger if l is a sensing fluent.

7 Conclusion

We believe our work on the specific properties addressed in this paper, in particular that related to improving goal achievement and reactive-awareness, is novel. However, our work complements the work of others in the field of formal analysis of agent systems. Amongst these are the following. Kacprzak et al [13] who have explored the use of unbounded model checking for verification of temporal epistemic properties. Lomuscio and Raimondi [12] have proposed a model checker called MCMAS that extends verification from temporal modalities to other modalities, such as correctness and cooperation, relevant to agents. Bordini et al [2] have used model checking for verification of properties of BDI agents expressed as AgentSpeak programs. Wooldridge et al [19] have presented a language called MABLE for multi-agent systems, allowing BDI-like agents and supporting automatic verification of properties via model checking.

We believe that the properties we have identified and discussed in this paper are interesting in general for all agent frameworks. Part of our future work is to study them in the context of other agent models.

Dunne et al [5] give computational complexity results for achievement and maintenance tasks of agents for a variety of environmental properties, for example whether or not the environment is deterministic, history dependent or bounded. It would be interesting to see whether some of our results could be strengthened for specific kinds of environments.

References

1. K. A. Arisha, F. Ozcan, R. Ross, V. S. Subrahmanian, T. Eiter, and S. Kraus. IMPACT: a platform for collaborating agents. *IEEE Intell. Systems*, 14(2), 1999.

2. R. H. Bordini, M. Fisher, W. Visser, and M. Wooldridge. Model checking rational agents. *IEEE Intell. Systems*, 19(5), 2004.
3. A. Bracciali, N. Demetriou, U. Endriss, A. Kakas, W. Lu, and K. Stathis. Crafting the mind of prosocs agents. *Applied Artificial Intelligence*, 20(2–4), 2006.
4. A. Bracciali, N. Demetriou, U. Endriss, A.C. Kakas, W. Lu, P. Mancarella, F. Sadri, K. Stathis, G. Terreni, and F. Toni. The KGP model of agency for global computing: Computational model and prototype implementation. In *Global Computing 2004 Workshop*, volume 3267 of *LNCS*. Springer Verlag, 2005.
5. P. E. Dunne, M. Laurence, and M. Woolridge. Complexity results for agent design problems. *Annals of Math., Computing and Teleinformatics*, 1(1), 2003.
6. K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J. Ch. Meyer. Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, 2(4), 1999.
7. A. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. Deliverable d4: A logic-based approach to model computees. Technical report, SOCS Consortium, 2003. Avaialble from `http://lia.deis.unibo.it/Research/Projects/SOCS/`.
8. A. C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. Declarative agent control. In *Post-proc. CLIMA V*, volume 3487 of *LNAI*. Springer Verlag, 2005.
9. A.C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. The KGP model of agency. In *Proc. ECAI-2004*, 2004.
10. R. A. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1), 1986.
11. J. A. Leite, J. J. Alferes, and L. M. Pereira. $\mathcal{MINERVA}$: A dynamic logic programming agent architecture. In *Proc. ATAL01*, volume 2333 of *LNAI*. Springer Verlag, 2002.
12. A. Lomuscio and F. Raimondi. MCMAS: a tool for verifying multi-agent systems. In *Proc. TACAS 2006*, volume 3920 of *LNCS*. Springer Verlag, 2006.
13. W. Penczek M. Kacprzak, A. Lomuscio. Verification of multiagent systems via unbounded model checking. In *Proc. AAMAS04*, 2004.
14. P. Mancarella, F. Sadri, K. Stathis, F. Toni, and A. Bracciali. Computees and welfare. Technical report, SOCS Consortium, 2005.
15. A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In *Proc. MAAMAW96*, volume 1038 of *LNCS*. Springer Verlag, 1996.
16. A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In *Readings in Agents*. Morgan Kaufmann Publishers, 1997.
17. F. Sadri, K. Stathis, and F. Toni. Normative KGP agents. *Computational and Mathematical Organization Theory*, 2006. To appear.
18. F. Sadri and F. Toni. Variety of behaviours through profiles in logic-based agents. In *Post-proc. CLIMA VI*, volume 3900 of *LNAI*. Springer Verlag, 2006.
19. M. Wooldridge, M.-P. Huget, M. Fisher, and S. Parsons. Model checking multi-agent systems: the mable language and its applications. *International Journal on Artificial Intelligence Tools*, 15(2), 2006.