This paper was selected by a process of
anonymous peer reviewing for presentation at

COMMONSENSE 2007

8th International Symposium on Logical Formalizations of Commonsense Reasoning

Part of the AAAI Spring Symposium Series, March 26-28 2007,
Stanford University, California

Further information, including follow-up notes for some of the
selected papers, can be found at:

www.ucl.ac.uk/commonsense07

Grounding Stanislavskian Scene Analysis in a Formal Theory of Action

Leora Morgenstern
IBM T.J. Watson Research Center
Hawthorne, NY 10532
leora@steam.stanford.edu; leora@us.ibm.com

Abstract

At the turn of the last century, Constantin Stanislavski developed a new system of acting, replacing the mannered gestures and forced emotion then popular with a more natural style. The core of his system lay in having actors perform a process of scene analysis, in which an actor would flesh out the circumstances of the play so that the character's motivations and actions would follow logically. This paper is an attempt to ground Stanislavski's method of scene analysis in a formal theory of action. We discuss the relations between Stanislavskian and formal AI theories of action and planning, give a formal definition of the end product of a scene analysis, and characterize the conditions under which a scene analysis is coherent.

1 Background and Motivation

Background: At the turn of the last century, Constantin Stanislavski, the founder of the Moscow Art Theatre, developed an innovative system of acting that broke with centuries of tradition in the theatre. Prior to Stanislavski, acting often relied heavily on stock mannerisms, such as putting one's hand to one's brow to indicate despair. A small minority of actors could express genuine emotion on the stage. But it was unclear, even to these actors themselves, how they achieved this display of emotion. An actor could work himself up into some emotional state during one performance, but might subsequently be unable to reproduce it. Still less could he teach others how to perform.

Stanislavski sought to develop a technique that could be taught and replicated. He was opposed to what he believed were the pitfalls of conventional acting: playing to the audience, conventional gestures, working oneself up into an emotional state. Instead he proposed that actors immerse themselves in the circumstances of the play. If an actor would sufficiently flesh out the circumstances of a play, he argued, he would be able to act in a realistic manner. The trick was the "magic if": hypothesizing enough facts, consistent with the play, to make it real to the actor, and enable him to feel, rather than pretend or artificially work-up, the appropriate emotions.

The cornerstone of Stanislavski's system [19] is the process of *scene analysis*. The actor constructs a backstory for his character, which includes a detailed portrait and history of the character before the start of the action of the play. and then chooses actions that further his objectives. If he does this exercise sufficiently well, according to Stanislavski, he can imagine in detail the circumstances of the play.

The aim of this paper is to explore the formalization of scene analysis, and characterize coherent scene analyses — the ones that seem to "work" for an actor.

Motivation: We list two motivations for this work:

First, there are striking similarities between the concerns of Stanislavski and of the formal AI/KR community; these are evident in the vocabulary and ontology used by Stanislavski and his followers. Common concepts include characters/agents, actions, objectives/goals, intention, causation, plans, and obstacles. Indeed, Stanislavski wrote about the need for a character's actions to follow *logically* from the circumstances that the actor has imagined; we are exploring to what extent we can develop a theory in which this happens.

Second, this is a promising domain for commonsense formalization. Much research in formal CSR has tended to focus on artificial problems, whether toy problems [12] or larger challenge problems [13; 17]. In contrast, the scene analysis problem is real. Actors and directors frequently use Stanislavski's methods, and have a good sense for when a scene analysis works. The process appears to rely more on commonsense reasoning than acting craft: even novice actors can do scene analysis. (Indeed, the fact that many people can perform scene analysis suggests that eventually, we may be able to evaluate formal theories of scene analysis by, say, having acting students rate scene analyses that are coherent according to some formal definition.)

Moreover, the scene analysis problem offers an interesting perspective on traditional domains in CSR research. For example, classical AI planning focusses on constructing plans in which success is guaranteed; while in a scene analysis, a character's plans may focus more on recovery from failure. That is, examining the scene analysis problem could lead to more realistic and commonsensical theories of planning.

2 The Scene Analysis Process

2.1 Example of scene analysis

There are many variations of the scene analysis process (SAP). We use here a modification of [4], itself based on [11] and [7]. The examples here refer to Stanislavski's production plan for *Othello* [20]. We focus on the first scene, which begins with Iago's urging Roderigo, who has unsuccessfully pursued the Venetian lady Desdemona, to tell her father, Brabantio, of her elopement with Othello.

The SAP includes the following steps:
1. Writing a *backstory* for one's character, including the personality traits of the character, and the actions that have happened prior to the start of the play
2. Determining a character's *scene objectives*
3. Determining which *strategies* the character uses to achieve his objectives, and the *actions* that each strategy comprises
4. Identifying the *obstacles* that stand in the way of the character executing his strategies
5. Choosing the *strategies* and *actions* that a character uses to overcome the obstacles.

For example, an actor playing Iago could do an (informal) scene analysis containing the following:
Iago's objective throughout the play: Avenging himself on Othello.
Iago's scene objective in the first scene: Breaking up Othello and Desdemona's marriage.
Strategy: Get Roderigo to inform Brabantio of the elopement so that Brabantio will ask officials to annul the marriage.
Obstacles: Roderigo is angry at Iago, doesn't trust Iago's judgement, and doesn't want to help him.
Strategy to get around obstacles: Appease Roderigo; convince Roderigo of his (Iago's) trustworthiness; remind Roderigo of how he has also been hurt by Othello; persuade Roderigo that informing Brabantio will indeed result in the desired outcome.

The actor could construct a backstory explaining why Roderigo is angry at Iago and doesn't trust his judgement. Stanislavski's extensive backstory recounts a growing friendship between Iago and Roderigo: Iago defends Roderigo from being beaten up by thugs who mock Roderigo for his pursuit of Desdemona; Iago offers to help Roderigo win Desdemona's hand; Iago asks Roderigo for a good deal of money in this pursuit; but at the start of the scene, Roderigo has discovered that Desdemona has eloped with someone else.

2.2 Scene analysis vs. classical AI planning

There are clear similarities between the planning implicit in scene analysis and classical AI planning. Nevertheless, there are some important differences:
First, the term *action* is used differently in the two contexts. In Stanislavskian scene analysis (SSA), when one refers to an *action*, one generally refers to a *dramatic action*, in the sense to be explained below. One can, e.g., refer to the (dramatic) actions of killing, convincing, or stealing. But there are many classical AI actions that are not considered actions in SSA. For example, Roderigo's utterance to Iago — "By heaven, I rather would have been his hangman" — is not considered an action. When Roderigo says these lines, he might be rejecting Iago's explanation, or sympathizing with Iago, depending on the actor's scene analysis. Rejecting an explanation or sympathizing are (dramatic) actions. The utterance of a line itself, however, is never an action. Similarly, movements across the stage (entrances, exits, crossing the stage) are not considered actions in SSA.

We can divide actions into 3 groups:
1.**locutionary actions**: the utterances (from the script) that the character speaks;
2. **blocking actions**: the movements that the character makes while on stage; [1]
3. **dramatic actions**: the essential actions that move a play forward and move each character toward his objectives. Examples: Iago's convincing Roderigo to ally himself with Iago, and Roderigo informing Brabantio of the elopement.

The SAP is primarily concerned with dramatic actions. Instances of locutionary and blocking actions may be co-extensional with instances of dramatic actions: i.e., one may perform a dramatic action *by performing* (in the sense of [5]) a locutionary and/or blocking action. For example, A might perform the action of *consoling* B by placing his arms around B (a blocking action) and uttering "there, there" (a locutionary action). An analysis of the relationship between locutionary and/or blocking actions, on the one hand, and dramatic actions on the other, is a very difficult problem and beyond the scope of this paper.

Indeed, analyzing the relationship between locutionary and dramatic actions — determining what is really happening when a character utters a line — is at least as difficult as the general story understanding problem. It is in fact *more* difficult because a story provides clues about tone and affect ("he said angrily") that are often absent in a script. The problem is still harder when one reasons about blocking actions. E.g., the dramatic action associated with the locutionary action of A uttering "there, there" to B can vary depending on whether the accompanying blocking action is A putting his arms around B or A casually tossing a box of tissues to B.
Second, the focus on planning is different. AI planning focusses on constructing a plan which is guaranteed to achieve the desired result. There is no such guarantee when characters construct plans; indeed, plays would be of limited interest if characters' plans always succeeded. Rather, a character does his best to choose actions which

[1] Blocking actions will be ignored in the rest of this paper. Directors generally tell actors to cross out all blocking-action stage directions before starting the SAP. Instead, blocking ought to follow naturally from one's dramatic actions, which vary with the director and actors of a particular production.

he believes will help him pursue his objectives, and replans as necessary.

Consider, e.g., Iago's objective to break up Othello and Desdemona's elopement. His strategy is to convince Roderigo to inform Brabantio, so that Brabantio will appeal to Venetian authorities to annul the marriage. Iago knows that a precondition of convincing Roderigo is having Roderigo favorably disposed toward him. He knows that if Brabantio would turn out to be unmoved by the elopement, then his strategy would fail. In fact, he knows when his strategy fails (for another reason, when the duke refuses to annul the marriage.) And he replans, choosing another strategy (making Othello insanely jealous of Desdemona) that will achieve his larger scene objective, namely, avenging himself on Othello.

Third, we note that in the AI planning community, there are two ways to represent goals, as *states* that the planning agent tries to achieve, and as *tasks* that the agent wishes to accomplish [16; 1]. The first approach is far more popular in the AI literature. In contrast, in SAP, an objective is typically spoken of as a verb or action: e.g., Iago's objectives are to *avenge* himself on Othello and*break up* Othello's and Desdemona's marriage. We find it awkward, however, to represent scene objectives as dramatic actions/tasks. Rather, it is more useful to identify objectives with goal states, but to allow natural language descriptions of objectives to be verb-based when this facilitates discussion. We use simple tricks to turn objectives that are more naturally associated with verbs into states: e.g., Roderigo's objective to marry Desdemona is represented as the state in which Roderigo is married to Desdemona.

3 A Formal Theory of Scene Analysis

Notation: The logic is sorted; all variables are universally quantified unless otherwise indicated. A list of all functions and predicates used, along with all definitions and axioms, is available at Appendix B [15].

3.1 Theory of Action

Our language is based on [2]. We use a situation-based branching theory of time. Intervals are defined by their starting and ending situations. Fluents are properties that change over time. *Holds(s,f)* (*Holds([s1,s2],f)*) indicates that fluent *f* holds over situation *s* (interval *[s1,s2]*).

Actions take place over intervals of time. *Occurs(ac,s1,s2)* denotes the action *ac* occurring between *s1* and *s2*. We can also say *Occurs(Do(a, act), s1, s2)*, which denotes the action of agent (or character) *a* performing *act* between *s1* and *s2*.*act* denotes an *actional*, an action which is not anchored to a particular agent.

The theory of knowledge and belief is based on the possible-worlds theory of knowledge introduced by [10] and extended by [14]. *B(a,s1,s2)* denotes that situation *s2* is indistinguishable to *a* from *s1*, given *a*'s beliefs.

Expected Effects, Success, and Failure

It will be useful to refer to the *preconditions*, *success conditions*, *failure conditions*, and *effects* of actions. The preconditions of an action are those fluents which must be true at the start of an action in order for an action to be performed. Effects and success conditions are closely connected to one another: an action has certain effects — fluents which hold at the end of an action — as long as certain success conditions (fluents) hold at the beginning of the action. Failure conditions are fluents which, if they hold at the beginning of an action occurrence, preclude the effects holding at the end of the action. Note that if one has a complete description of all success conditions, one can derive the failure conditions. Agents, however, typically don't have such complete descriptions. Nevertheless, an agent is typically aware of at least some of the failure conditions for the actions that he intends to perform, and if he becomes aware that a failure condition holds, may work to change that fluent's truth value or choose another action to perform.

Any action theory in this domain will therefore include three types of axioms:

Precondition axioms, of the form *Occurs(ac, s1, s2)* $\Rightarrow$ *Holds(s1, f)*. *f* is a precondition of *ac*.

Effect axioms, of the form *(Occurs(ac, s1, s2)* $\wedge$ *Holds(s1,f1))* $\Rightarrow$ *Holds(s2, f2)*. *f1* is a success condition of *ac*, and *f2* is the effect.

Failure axioms, of the form *(Occurs(ac, s1, s2)* $\wedge$ *Holds(s2, f2))* $\Rightarrow \neg$ *Holds(s1,f1)*. *f1* is a failure condition for *ac*.

We introduce the following predicates on actions:

- *Precond(ac,f)*: *f* is a precondition of performing *ac*
- *SuccessCond(ac,f1,f2)*: *f1* is a sufficient condition for the successful performance of *ac*, resulting in effect *f2*
- *FailCond(ac,f1,f2)*: *f1* is a sufficient condition for the failure of *ac* to achieve *f2*.

The first and third predicates will be useful in stating the definition of coherence in Section 3.3 and in stating the axioms and developing the proof in Appendix A [15].

Objectives and Strategies

Objectives: An objective is represented as a fluent, a state that an agent wants to achieve.

Strategies: Our development of strategies is influenced by the ideas of [6]. A strategy is a relatively loose structure of actions, much like a partial plan, with the following characteristics:

- It comprises a set of actions, or other strategies, that may be used to accomplish a particular strategy.
- It may, but does not necessarily, mandate the order in which these actions/strategies must be performed/executed.
- It does not necessarily contain the complete set of actions necessary for success.
- Its execution does not guarantee success in achieving one's objective.
- It will typically have gaps; there is not an action prescribed for every time period in which the strategy is executed. (An agent may execute a second strategy during a gap in the first strategy.)
- It may include actions performed by other agents. E.g., Iago's strategy for breaking up Othello and Desdemona's

marriage includes Roderigo's action of informing Brabantio of Othello and Desdemona's elopement. (The strategy will, of course, fail if the other agent does not do his action in a timely manner.)

• It may include *reactive actions* — actions that respond to a particular situation or action of another agent. E.g., (Section 4), a strategy to keep a secret may include the reactive action of refusing to answer an agent's question if doing so would entail the secret becoming known.

Analogous to the concept of an action occurring during an interval is the notion of a strategy being *executed* during that interval. We distinguish between a complete and an incomplete execution. Intuitively, a strategy *strat* is completely executed during an interval if all non-conditional actions included in *strat* occur or, recursively, if all non-conditional strategies included in *strat* are executed; if all conditional actions/strategies occur/are executed if the conditions hold; and if the order in which these actions occur and strategies are executed satisfy the temporal constraints. If a strategy is begun but not concluded (for whatever reason, e.g., if some action in the strategy is not executed), it is said to be incompletely executed. The predicate *Executes(a,strat,f,s1,s2)* denotes that *a* executes strategy *strat* in pursuit of objective *f* between *s1* and *s2* and that the execution is completed.*StartExecute(a,strat,f,s1,s2)* denotes that *a* begins to execute the strategy *strat*, and that the execution takes place between *s1* and *s2*. The execution may not be complete at *s2*.

To define a strategy, we first introduce some syntactic sugaring conventions.

• $Do(a,\ Act(\vec{x})\ |\ P(\vec{x}))$ denotes the action of *a* doing actional *Act* with the range restricted to P in the obvious way.

• In general, $Occurs(Do(a,\ Act(\vec{x})\ |\ P(\vec{x})),\ s1,s2)$
$\Leftrightarrow \exists$x (P($\vec{x}$) $\wedge Occurs(Do(a,\ Act(\vec{x})),\ s1,\ s2))$.

We call a sentence of the form *Occurs(ac,si,sj)* or of the form above, or either of these forms preceded by a negation sign, an *occurrence sentence*.

These syntactic sugaring conventions are extended in an analogous way to the *Executes* and *StartExecute* predicates.

We call a sentence of the form *Executes(a, strat, f, si, sj)*, *StartExecute(a, strat, f, si, sj)*, the syntactically sugared form, or any of these forms preceded by a negation sign, an *execution sentence*.

For an occurrence or execution sentence, the *active agent* is defined as the first argument to the *Do* function or the *Execute/ StartExecute* predicates; the *active interval* is defined as the interval formed by the last two arguments of the *Occurs*, *Execute*, or *StartExecute* predicates.

Strategies, like actions, can be viewed as sets of intervals. A strategy is of the form {(ss,se) | $\Sigma \wedge \kappa$}, where Σ is a conjunction of action and strategy occurrence formulas, κ is a conjunction of temporal constraints, and Σ and κ satisfy the following:

• Each conjunct of Σ is of the form
$\Psi_1(\vec{x}) \ldots \Psi_m(\vec{x}) \Rightarrow \Phi_1(\vec{x}) \ldots \Phi_n(\vec{x})$, where

(a) $\vec{x}$ represents an array of variables, including situational, action, and agent variables

(b) any of the Ψ_i or Φ_i is an occurrence or execution sentence as defined above, or of the form [¬] *Holds(s,f)* and

(c) at least one of the Φ_i is an occurrence or execution sentence.

Each conjunct of κ is of the form $s_i < s_j$ or $s_i \leq s_j$, for some *i,j*.

Example: {(s1,s4) | Occurs(Do(a1,Act1(x) | P1(x)), s1,s2) ∧ Occurs(Do(a2,Act2(x) | P2(x)), s3,s4) ∧ s2 ≤ s3 }
is an example of a strategy. More examples can be seen in Appendix A.

Interaction between strategies and objectives

At any situation in time, an agent has at least one primary objective, his scene objective. Strategies are used to achieve objectives; the pursuit of a strategy may generate other objectives. As an agent uses a strategy to achieve his objectives, he may form objectives to achieve preconditions or avoid failure conditions for the actions in his strategy. In order to formalize the interaction between strategies and objectives, we need to express how an agent proceeds through his strategy. The following functions and predicates will facilitate this discussion:

• *ActionOf(strat, ac)* denotes that *ac* is one of the actions in strategy *strat*.

• *Precursor(ac1,ac2,strat)* denotes that *ac1* must be performed prior to *ac2* when *strat* is executed. (This relation is entailed by the temporal constraints in a strategy.)

• *StrategyFor(f,strat)* denotes that strategy *strat* is a strategy for pursuing objective *f*.

• *Holds(s,SObj(a,f))* denotes that *f* is the scene objective of *a* in *s*.

• *Holds(s,CObj(a,f))* means that *f* is a current objective of *a* in *s*. (There may be multiple objectives.)

• *Holds(s,CStrat(a,f,strat))* denotes that *strat* is *a*'s current strategy in *s* in pursuit of his objective *f*.

• *Holds(s, CAction(strat, a1, f, do(a2,act)))* denotes that the action *do(a2,act)* is a current action for *a1*'s strategy *strat* to achieve objective *f*. Note that *a1* may be distinct from *a2*.

An action *ac* is said to be *done* in *s* relative to some agent *a* and strategy *strat* if there was some interval, ending in *s*, in which *strat* was the current strategy of *a* for achieving objective *f*, and *ac* occurred at some point during that interval.

Definition of done:
Holds(s, Done(ac,a, strat)) ⇔
ActionOf(ac,strat) ∧
∃ *f, ss', ss, sa, sb Holds([ss,s], CStrat(a,f,strat))* ∧
∀ *s' ss'* $\leq s' < s \Rightarrow \neg$ *Holds(s',CStrat(a,f,strat))* ∧
sa $\geq$ *ss* ∧ *sb* $\leq$ *s* ∧ *Occurs(ac, sa, sb)*

An actional *act* is a *potential action* for an agent *a* pursuing some strategy *strat* if all the precursors of *ac* in *strat* have already been done:

Definition of potential action:
Holds(s,PotAct(a,act,strat)) ⇔
∀ *ac Precursor(ac,Do(a,act),strat)*
⇒ *Holds(s,Done(ac,a,strat))*

Strategy Failure: The notion of strategy failure is central to the development of a formal theory of scene analysis. While an agent is not required to predict the success of his strategies — indeed, because he has no such requirement — he needs to realize when his strategies are not working out. *Holds(s, StrategyFailed(a,f,strat))* says that at situation *s*, the particular strategy *strat* that agent *a* has chosen in his pursuit of objective *f* has failed.

Although a complete analysis of strategy failure is beyond the scope of this paper, we make some observations about the circumstances in which a strategy may fail:
1. *a* is pursuing a strategy and has performed an action in this strategy, but the expected effect does not hold.
2. *a* is pursuing a strategy, which calls for him to perform an action. The action has a precondition which does not hold, and he either does not know or cannot perform an action to establish that precondition.
3. *a* is pursuing a strategy, which calls for another agent *a'* to perform an action. *a'* performs the action, but the expected effect does not hold.
4. *a* is pursuing a strategy, which calls for another agent *a'* to perform an action. *a'* does not perform the action, or performs the contrary of the action.

An example of the fourth type of failure can be seen in Iago's strategy to break up Desdemona's and Othello's marriage. His strategy consists of his convincing Roderigo to inform Brabantio of the elopement; for Brabantio to alert the duke; and for the duke to annul the marriage. However, the duke does not annul the marriage; in fact, he confirms that it is valid.

When a strategy fails, an agent may repeat the strategy or choose another strategy to achieve his objective. E.g., when Iago's initial strategy to break up Desdemona and Othello fails, he chooses another strategy: making Othello jealous of Desdemona. It is not trivial to characterize in what circumstances an agent will switch strategies or repeat a strategy/action. Certainly, plays — and life — are rife with examples of agents who persist in a strategy and prevail. Iago, for example, must repeatedly entreat Roderigo before the latter agrees to inform Brabantio of the elopement. Yet unrestrained persistence is, at best, the stuff of slapstick comedy.

An analysis of strategy persistence vs. strategy switching might formalize the following: that agents may persist in a strategy for a certain amount of time, or repeat an action several times until they reach some threshold of tolerance; that the threshold that an agent has for repetition may depend on a variety of factors, including the ease of performing an action, expected payoff, availability of other strategies, or difficulty of executing such other strategies. This is left for future work.

Motivation

One wishes not merely to posit an agent's scene objective, but to ground this objective. The backstory can provide this grounding. For example, Stanislavski's backstory for *Othello* [20] explains the past connection between Othello and Iago, detailing occasions where Iago saved Othello's life; Iago's low-born background; Othello's decision to choose as lieutenant the high-born but unworthy Cassio, because he needs to appear polished in elegant Venetian society. This *motivates* Iago's resentment and explains why Iago wants to avenge himself on Othello.

The notion of motivation used here is significantly weaker than that, say, of [21] (where an action was motivated if its occurrence was entailed). This theory retains the concept of free will: No matter what has happened, a person is never forced to choose an objective. Rather, we introduce the predicate *Motivated(a, f)*, provide axioms for this predicate, and then show that particular backstories entail particular instantiations of the *Motivated* predicate. Even if an objective is motivated for a particular character, however, it is not necessarily the character's scene (or current) objective.

3.2 Scenes and Scene Analysis

We define a scene *SC* as a tuple $< Char, \Sigma >$, where *Char* is the set of agents/characters in the scene and Σ is a sequence of (mostly) locutionary actions. (Σ may include dramatic actions that are forced (entailed) by the script. E.g., the script of *Othello* entails that Othello kills Desdemona. However, in general, most dramatic actions are introduced during the SAP.)

We define a scene analysis *SA(SC, A')* as a tuple $< Char, \Sigma$, *[SS,SE], BStory(A',SS), Obj,* Δ*(A',SS,SE),* $\Pi >$, where
• *Char* and Σ are as above
• *SS* and *SE* are the starting and ending situations of this instantiation of the scene,
• *BStory(A',SS)* is the backstory of character *A'* up to situation *SS*, defined as a set of sentences, each of which is of the form *Holds(s, f)* where $s \leq SS$, or is an occurrence sentence whose latest time point is earlier than *SS* and whose active agent is in *Char*
• *Obj* is a set of fluents, the objectives of *A'*,
• Δ is the dramatic history of the scene, defined as a set of sentences each of which is of the form *Holds(s,f)* or is an occurrence/execution sentence whose active agent is in *Char* and whose active interval is contained in the interval *[SS,SE]*
• Π relates subsets of Δ to subsets of Σ. That is, Π associates dramatic actions with lines in the script. In general, one line of the script may be associated with several dramatic actions, and one dramatic action may be associated with several lines in the script.

Let Γ*(SA(SC,A'))* be the union of the sentences in the backstory and the dramatic history.

3.3 Coherence

Our goal is to characterize those scene analyses that make sense, that "work" for an actor. Informally, we would like to say that a scene analysis is *coherent* if the following conditions hold:
[1] The scene objectives are motivated with respect to the backstory
[2] Any other objectives arise from the original scene objectives, the strategies taken to pursue objectives, and

the facts that are true during the scene
[3] An agent will pursue a strategy only for an objective
[4] An agent's actions during the scene follow from his objectives and chosen strategies
[5] An agent will not continue a strategy that he believes has failed.

Definition: Let SC be a scene and SA(SC,A') a scene analysis for character A', as defined above. Let Γ*(SA(SC,A'))* be the set of wffs associated with the scene analysis, as defined above. Let $\Gamma(CSK)$ be a set of sentences representing a body of commonsense knowledge. (E.g., for *Othello*, this might include commonsense domain theories about wooing spouses, and fathers' reactions to their daughters' elopements.)

Then SA is coherent iff $\Gamma(SA(SC,A')) \cup \Gamma(CSK) \models$

1. (motivation of scene objectives)
$(\forall s \in [ss, se]\ Holds(s,SObj(A',f)) \Rightarrow Holds(s, Motivated(A',f)) \wedge$

2. (subgeneration of other objectives)
$(Holds(s, CObj(A',f)) \Rightarrow Holds(s, SObj(a,f)) \vee$
$\exists\ strat, ac, f'\ (Holds(s,CStrat(A',f, strat)) \wedge ActionOf(ac,strat) \wedge \neg Holds(s, Done(ac,A', strat)) \wedge (Precond(ac,f) \vee FailCond(ac,\neg f,f'))) \wedge$

3. (strategy pursuit only for objectives, and only if not failed)
$(Holds(s,CStrat(A',f, strat)) \Rightarrow$
$((Holds(s,CObj(A',f)) \wedge StrategyFor(strat,f) \vee$
$(Holds(s,CObj(A',f') \wedge StrategyFor(strat',f')$
$\wedge StratPart(strat,strat')) \wedge$
$\neg\ (B(A',s,s') \Rightarrow Holds(s',StrategyFailed(A',f,strat))))$
$\wedge$

4. (actions are performed by A' only if done as part of some strategy and only if it is believed that they will not fail)
$(occurs(s,s',do(A',act)) \Rightarrow$
$Holds(s,CStrat(A', f,strat)) \wedge$
$Holds(s,PotAct(A',act,strat)) \wedge$
$\neg\exists\ f'\ (FailCond(Do(A',act,f,f') \wedge \forall s(B(A',s,s') \Rightarrow Holds(s',f)))$

4 Example

To demonstrate how one can perform a scene analysis and show that it is coherent, we use a small sample script, used in teaching principles of scene analysis, adapted from [9].

A: Give me that.
B: No.
A: Give it to me.
B: I don't think so.
A: Come on: I really want it.
B: No!
(A grabs it from B.)
B: Well?
A: Well what?
B: Well, say something.
A: What do you want me to say?
B: You might have something to say.
A: I'm not going to say anything.

This mini-scene is clearly ambiguous (are the characters two children arguing in the playground over a toy? a parent forbidding something to a child?); the point of scene analysis is fleshing it out.

What follows is an overview. (The full analysis is in Appendix A [15].) We do the scene analysis from B's point of view. First we present the scene analysis: We give the scene (characters and set of locutionary actions), posit a backstory, B's objectives, the dramatic history, and the mapping between locutionary actions and the dramatic history. Then we prove that the scene analysis is coherent according to the definition in Section 3.3.

We note the following:

- There is no attempt to automate construction of a backstory or dramatic history. The aim is not to automate creative analysis, but to demonstrate that a particular example of creative analysis is coherent.
- There is no attempt to represent the locutionary actions as anything beyond *Do(a,utter(q))*, where *q* is a string of the English language. As argued in Section 2.2, demonstrating the relationship between locutionary actions and dramatic actions is a very difficult problem and beyond the scope of this paper.

We highlight the main points of the scene analysis:

The backstory: We posit a backstory where A and B are in a relationship. B wants to break up with A. B has already purchased a one-way ticket to the Bahamas. A has just seen that B is holding something, but doesn't know what it is. B is a non-confrontational person.

B's objective: breaking up with A without having to say to A's face that she wishes to break up with him.
B's strategy is the Runaway strategy. It consists of hiding her desire to break up with A until she begins her trip, taking her trip, and then writing a letter to inform A of the breakup. Hiding something is itself a strategy. It consists of a reactive action: if Y's objective is to hide something, and X asks Y to do something which would entail X's finding out, then Y must refuse.

The dramatic history: A asks B to hand him the ticket. B knows that if she agrees to this request, A will find out that she has a one-way ticket, and will infer that she is planning a breakup. To execute the hiding strategy, B must therefore refuse A's request. This is repeated three times.

A then grabs B's ticket. At this point, B's strategy to hide her desire to break up with A has failed. Indeed, her Runaway strategy has failed. However, she still has the same objective: to get out of her relationship with A. She now switches strategies, to taunt A with the fact that she has a one-way ticket to the Bahamas. She tries this several times, but A does not take the bait.

The proof of coherence: The formal proof is simple and consists mostly of matching definitions. First (condition 1), we demonstrate that B's objective is motivated by the backstory. This follows from some commonsense domain axioms on relationships and non-confrontational tendencies.

Next (condition 2), we demonstrate that at any point, all current objectives of B are scene objectives or are generated from the scene objectives. Since we only deal with B's single scene objective, this is trivial.

Next (condition 3), we demonstrate that B pursues her current strategy only when it lines up with her current objective and only when she knows the strategy hasn't failed. Assume that A's grabbing the ticket occurs between *Sa* and *Sb*. Until *Sb*, B's current strategy is the Runaway strategy. At *Sb*, B realizes that this strategy has failed, and switches to the Taunt strategy/action, which still lines up with her current objective.

Next (condition 4), we demonstrate that B performs actions only if they are part of her current strategies. We consider each of B's 6 actions. Her first 3 actions, which happen before *Sb*, are refusing A's request to hand over the ticket. These are part of the Hiding strategy, which is itself part of the Runaway strategy. Her next 3 actions, which happen after *Sb*, are part of the Taunt strategy/action.

This completes the informal discussion of the proof.

5 Related Work

There have been two previous studies relating AI and Stanislavskian theory. El-Nasr [3] develops an interactive narrative architecture, based on certain aspects of Stanislavskian theory, and uses it for various virtual entertainment applications. She uses a version of a scene analysis ontology as the basis of her application's data structure. Hoffman [8] has considered how one might apply Stanislavskian theory to construct robots that interact with humans. A primary focus is the physical actions and gestures that a robot would perform. Neither El-Nasr nor Hoffman works in formal logic, and neither addresses the notion of coherence.

There are clear connections between our work and first, the long tradition, dating back to [18], of story understanding using knowledge of an agent's goals and plans; and second, the work toward a declarative theory of reactive planning [22]. These lines of research are less formal than our theory, and do not address the notion of coherence.

6 Conclusions and Future Work

We have presented a formal theory of Stanislavskian scene analysis. Our theory builds upon previous theories of action, but adds several new elements, including concepts of strategy and strategy failure.

Future work includes first, developing a more detailed theory of strategy failure, by formalizing the discussion of section 3; and second, extending the notion of scene analysis to multiple agents. The question arises, when looking at the scene analyses of different characters in a scene, whether these scene analyses are consistent, and what level of inconsistency we can tolerate. One character need not be aware of all the beliefs or even actions of another character, but at some level, they must share beliefs and knowledge of what is happening in the scene.

In the long term, we wish to attempt to formalize a later stage of the SAP: determining which of a character's emotions are supported by a scene analysis. We believe that this requires much preliminary work in developing a formal structure for representing and reasoning about emotions.

Acknowledgements: I am grateful to Laurence Gewirtz for guidance and helpful discussions, and to Ernie Davis for his comments on an earlier draft of this paper.

References

[1] Eugene Charniak and Drew McDermott. *Introduction to Artificial Intelligence.* Addison Wesley, 1985.

[2] Ernest Davis. Knowledge and communication: A first-order theory. *Artificial Intelligence*, 166:81–140, 2004.

[3] Magy Seif El-Nasr. A user-centric adaptive story architecture: borrowing from acting theories. In *Advances in Computer Entertainment Technology*, pages 109–116. ACM, 2004.

[4] Laurence Gewirtz. Ten questions actors must consider when working on roles. unpublished, 2005.

[5] Alvin I. Goldman. *A Theory of Human Action.* Prentice-Hall, Englewood Cliffs, 1970.

[6] Andrew S. Gordon. The representation of planning strategies. *Artif. Intell.*, 153(1-2):287–305, 2004.

[7] Uta Hagen and Haskel Frankel. *Respect for Acting.* Macmillan, New York, 1973.

[8] Guy Hoffman. Acting lessons for artificial intelligence. In *Proc., 50th anniv. summit of AI (AI-50).* 2006. http://web.media.mit.edu/ guy/publications/AI-50hoffman.pdf.

[9] Stanley Kahan and Kenneth W. Rugg. *Introduction to Acting.* Allyn and Bacon, 1998.

[10] Saul Kripke. A semantical analysis of modal logic i: Normal modal propositional calculi. *Zeitschrift fur Mathematische Logic und Grundlagen der Mathematik*, 9:67–97, 1963.

[11] Robert Lewis. *Advice to the Players.* Theatre Communications Group, 1991.

[12] John L. McCarthy. Elaboration tolerance. In *Working Papers of the Fourth International Symposium on Logical Formalizations of Commonsense Reasoning, Common Sense 98*, 1998.

[13] Rob Miller and Leora Morgenstern. The commonsense problem page, 1997. http://www-formal.stanford.edu/leora/commonsense.

[14] Robert C. Moore. Reasoning about knowledge and action, 1980. SRI Technical Report 191.

[15] Leora Morgenstern. Appendices a and b to grounding stanislavskian scene analysis in a formal theory of action, 2007. Available at http://www-formal.stanford.edu/leora/st and at http://www.ucl.ac.uk/commonsense07/papers/notes/morgenstern.

[16] Earl D. Sacerdoti. *A Structure for Plans and Behavior.* Elsevier North-Holland, New York, 1977.

[17] Erik Sandewall. *Logic Modelling Workshop.* 1999. http://www.ida.liu.se/ext/eta/lmw.

[18] Roger Schank and Robert Abelson. *Scripts, Plans, Goals, and Understanding.* Lawrence Erlbaum Associates, 1977.

[19] Constantin Stanislavski. *An Actor Prepares.* Routledge, New York, 1936.

[20] Constantin Stanislavski. From the production plan of othello. In Toby Cole, editor, *Acting: A Handbook of the Stanislavski Method, revised edition.* Three Rivers Press, New York, 1983.

[21] Lynn Andrea Stein and Leora Morgenstern. Motivated action theory: a formal theory of causal reasoning. *Artificial Intelligence*, 71(1):1–42, 1994.

[22] P. Traverso, E. Giunchiglia, L. Spalazzo, and F. Giunchiglia. Formal theories for reactive planning systems. In *AAAI'96 Workshop on Theories of Action, Planning, and Control.* 1996.