Modeling Mobile Intention Recognition Problems with Spatially Constrained Tree-Adjoining Grammars

Peter Kiefer
Laboratory for Semantic Information Technologies
University of Bamberg
peter.kiefer@uni-bamberg.de

Abstract

The problem of inferring an agent's intentions from her spatio-temporal behavior is called *mobile intention recognition problem*. Using formal grammars we can state these problems as parsing problems. We argue that context-free formalisms are not sufficient for important use cases. We introduce *Spatially Constrained Tree-Adjoining Grammars* which enrich TAGs with knowledge about the structure of space, and about how space and intentions are connected.

1 Introduction

The interaction possibilities between a mobile user and her device are often restricted. Mobile usage scenarios, such as navigation (Krüger et al., 2004), location-based gaming (Schlieder et al., 2006), and maintenance work (Kortuem et al., 1999), imply that the user's haptic and cognitive resources (Baus et al., 2002) are bound by a specific task. In these situations we would desire a system that somehow 'guesses' the user's information needs and presents the information automatically. The system must have a complex model of the intentions that are possible in a specific use case, and find an intention which consistently explains the user's behavior (*intention recognition problem*, IR).

In literature, the IR problem is also known as *plan recognition (PR) problem* (Carberry, 2001). It can be seen as the problem of revealing the hidden structural regularities that underlie an agent's sequence of behaviors. Formal grammars are often used to describe structural regularities, not only in natural language processing (NLP), but also in areas like computer vision (Chanda and Dellaert, 2004), and action recognition (Bobick and Ivanov, 1998). Consequently, formal grammars were also considered for PR/IR (Pynadath, 1999). Recent work has drawn parallels between NLP and PR/IR and argued that the expressiveness of context-free grammars (CFG) is not sufficient for important use cases (Geib and Steedman, 2007; Kiefer and Schlieder, 2007).

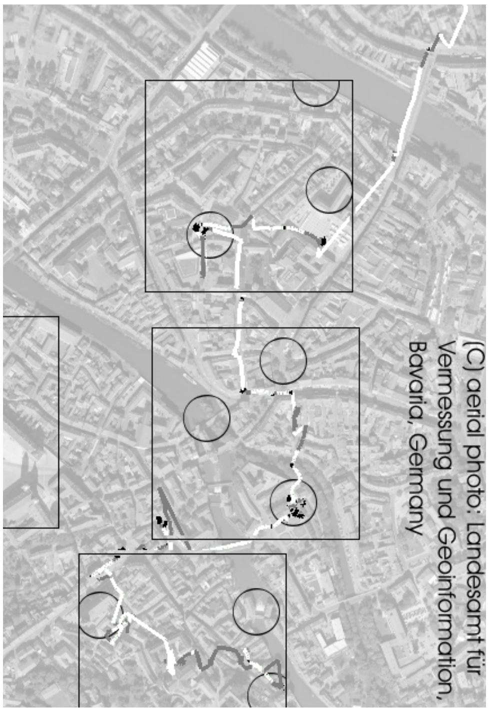

(C) aerial photo: Landesamt für Vermessung und Geoinformation, Bavaria, Germany

Figure 1: Spatio-temporal behavior in a location-based game: what are the user's intentions?

This paper continues this line of research by proposing mobile IR problems as an application area for Tree-Adjoining Grammars (TAG). We first explain which steps are necessary to state a mobile IR problem as parsing problem (section 2). We make the point that this problem class is special because mobile behavior happens in space and time. Section 3 introduces *Spatially Constrained Tree-Adjoining Grammars* (SCTAG), and explains how they can be used to model complex intention-space-relations elegantly. We close with an overview on related work (section 4) and an outlook on our future research (section 5).

2 Mobile intention recognition with formal grammars

2.1 Bridging the gap

One implication of spatio-temporality is that the gap between sensor input (e.g. position data from a GPS device) and high-level intentions (e.g. *'find a restaurant'*) is extremely large. To bridge this gap, we use a multi-level architecture with the level of *behaviors* as intermediate level between *position* and *intention*. We process a stream of (lat/lon)-pairs as follows:

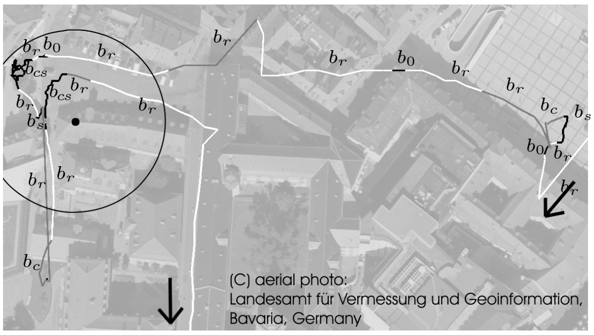
1. *Preprocessing* The quality of the raw GPS data is improved. This includes removing points with zero satellites, and those with an impossible speed.

2. *Segmentation* The motion track is segmented at the border of regions, and when the spatio-temporal properties (e.g. speed, direction) of the last *n* points have changed significantly (Stein and Schlieder, 2005).

3. *Feature Extraction* Each segment is analyzed and annotated with certain features, like speed and curvature (Schlieder and Werner, 2003).

4. *Classification* Using these features, each motion segment is classified to one *behavior*. We can use any mapping function from feature vector to behaviors, for instance realized as a decision tree.

As output we get a stream of behaviors. In the example from Fig. 2 we distinguish the following spatio-temporal behaviors: riding (b_r), standing (b_0), sauntering (b_s), curving (b_c), and slow-curving (b_{cs}). In other use cases we might as well have non spatio-temporal behaviors, like manual user input. We call an IR problem a *mobile* one if at least some behaviors are spatio-temporal ones.



(C) aerial photo:
Landesamt für Vermessung und Geoinformation,
Bavaria, Germany

Figure 2: Spatio-temporal behavior sequence in the first region.

The track from Figures 1 and 2 was recorded in the location-based game CityPoker. In the scope of this paper, we will only roughly introduce the rules of this game whenever needed. For a complete description, refer to (Schlieder, 2005). The reason why this game is especially suited as exemplary use case is that CityPoker is played by bike at high speed.

2.2 Parsing behavior sequences

The stream of behaviors described above serves as input to a parsing algorithm. Using behaviors as terminals and intentions as non-terminals, we can write rules of a formal grammar that describe the intentions of an agent in our domain. Most plan recognition approaches have followed a hierarchical structure of plans/intentions (e.g. (Kautz and Allen, 1986; Geib and Goldman, 2003)). In CityPoker, for instance, a player will certainly have the intention to *Play*. At the beginning of each game, the members of a team discuss their strategy. Playing in CityPoker means exchanging cards in several cache regions, so we model a sequence of intentions as follows: $GotoRegion$ $HandleRegion$, $GotoRegion$ $HandleRegion$, and so on. In the cache region players find themselves a comfortable place to stand, answer a multiple-choice question, and select one out of three caches, depending on their answer. In the cache, they search a playing card which is hidden in the environment (see the behavior sequence in Fig. 2).

A context-free production system for CityPoker is listed in Fig. 3[1]. The choice of the formalism de-

[1] Rules with a right-hand side of the form $(symbol_1|...|symbol_n)^+$ are a simplified notation for 'an arbitrary sequence of $symbol_1, ..., symbol_n$, but at least one of them'.

pends on the requirements of the use case. As argued in (Schlieder, 2005), most intention recognition use cases need at least the expressiveness of a CFG. A typical example is leaving the same number of regions as entered before ($enter^n leave^n$). We can find the currently active intention in the parse tree by choosing the non-terminal which is direct parent of the current behavior.

2.3 Spatially Grounded Intentional Systems

Up to here we have largely ignored the spatial aspect of mobile intention recognition. We have used space in the preprocessing, but the last subsection was nothing but a simple CFG with intentions and behaviors. Now we will see how space can help us to reduce ambiguity. Consider the two parse trees in Fig. 4: both are possible for the behavior sequence from Fig. 2. In the upper one the agent has entered the circular cache and is searching for the cards. In the bottom one the agent is in the region and still searching for the cache. Obviously, the upper one can only occur if the behaviors are located in a cache. This is the basic idea of *Spatially Grounded Intentional Systems* (SGIS) (Schlieder, 2005): SGIS are context-free production systems with the extension that each rule is annotated with a number of regions in which it is applicable. We call this the *spatial grounding* of rules. For instance, a $HandleCache$ intention is grounded in all regions of type $cache$. We modify all rules accordingly. An SGIS rule for the original rule (12) would look like follows:

$$\begin{aligned} HandleCache \quad &\rightarrow \\ SearchCards \quad & DiscussStrategy \\ [grounding: cache_{1,1}, &..., cache_{5,3}] \end{aligned}$$

This reduces the number of possible rules applicable at each position in the behavior sequence, thus avoiding many ambiguities. For parsing in SGIS we replace the pure behavior stream $(beh_1, beh_2, beh_3, ...)$ by a stream of behavior/region pairs: $((beh_1, reg_1)$, (beh_2, reg_2), (beh_3, reg_3), ...). Each behavior is annotated with the region in which it occurs. Also the non-terminals in the parse tree are annotated with a region *(Intention, region)*, with the meaning that all child-intentions or child-behaviors of this intention must occur in that region. SGIS are a short form of writing rules of the following form (where $Symbol$ can be an intention or a behavior):

$$\begin{aligned} (Intention, reg_x) \quad &\rightarrow \\ (Symbol_1, reg_x) \quad ... \quad &(Symbol_n, reg_x) \end{aligned}$$

That means, we cannot write rules for arbitrary combinations of regions. In addition, we require that another rule can only be inserted at an intention $Symbol_i$ if the region of the other rule is (transitive) child in the partonomy, i.e. in the above rule we can only insert productions with a region $reg_y\ part\_of\ reg_x$ (which includes the same region: reg_y.equals(reg_x)). SGIS have been designed for partonomially structured space. The nesting of rules follows closely the nesting of regions and sub-regions in the spatial model. The CityPoker partonomy is structured as follows: the game_area contains five rectangular cache_regions, each of which in turn contains three caches (see Fig. 1.

3 A 'Spatialized' TAG

3.1 Spatial constraints

SGIS support a partonomial structure between regions, i.e. only *part_of* relations exist. In general, a lot more topological relations are possible, like *touches*, *disjunct*, *identical*, or *north-of*. Examples can be found in the literature on geographic information science (Egenhofer and Franzosa, 1991). This restriction of SGIS hinders us from expressing frequently occurring use cases. Consider the motion track in Fig. 2: the agent enters the cache, shows some searching behavior, and then temporarily leaves the circular cache to the south. Knowing the whole motion track we can decide that this is an *AccidentalLeave* intention, and not a *ChangePlan* intention[2]. It is not necessary that the intermediate intention, let us call it $Confused$, is located in the parent cache_region of the cache. Finally, entering just *any* cache is not sufficient for an $AccidentalLeave$ intention, but we require that cache to be the same as left before.

[2] A player in CityPoker who has given a wrong answer to the quiz will be searching at the wrong cache and probably give up after some time. He will then head for one of the other caches. The $ChangePlan$ intention was omitted in Fig. 3 for reasons of clarity.

Production Rules for CityPoker			
Play	→	*DiscussStrategy Continue*	(1)
DiscussStrategy	→	b_0	(2)
Continue	→	ε | *GotoRegion HandleRegion Continue*	(3)
GotoRegion	→	$(b_r\|b_0\|b_c)^+$	(4)
HandleRegion	→	*SelectCache GotoCache HandleCache*	(5)
SelectCache	→	*FindParkingPos AnswerQuiz*	(6)
FindParkingPos	→	$(b_r\|b_c\|b_{cs})^+$	(7)
AnswerQuiz	→	b_0	(8)
GotoCache	→	(*SearchWayToC* |*NavigateTowardsC*)$^+$	(9)
SearchWayToC	→	$(b_0\|b_{cs}\|b_s)^+$	(10)
NavigateTowardsC	→	$(b_r\|b_c)^+$	(11)
HandleCache	→	*SearchCards DiscussStrategy*	(12)
SearchCards	→	(*CrossCache*|*DetailSearch*)$^+$	(13)
CrossCache	→	$(b_r)^+$	(14)
DetailSearch	→	$(b_0\|b_{cs}\|b_s\|b_c)^+$	(15)

Figure 3: Context-free production rules for intention recognition in CityPoker.

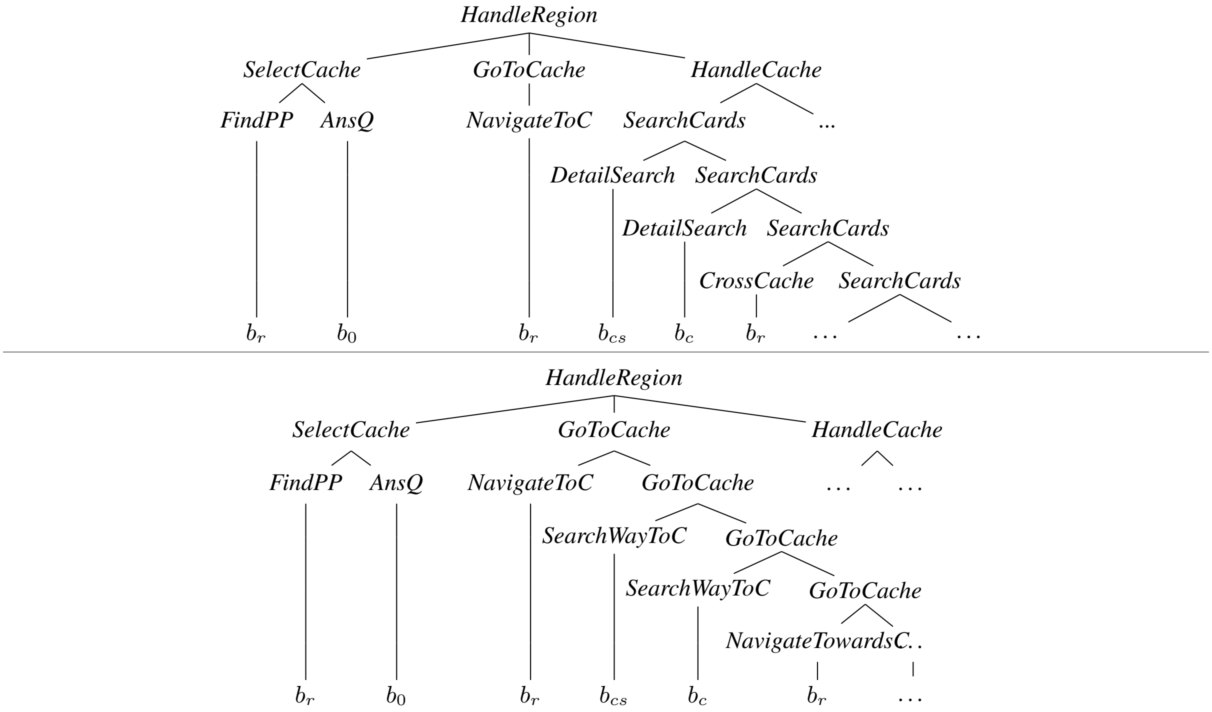


Figure 4: Parsing ambiguity if we had no spatial knowledge (see track from Fig. 2). Through spatial disambiguation in SGIS we can decide that the bottom parse tree is correct.

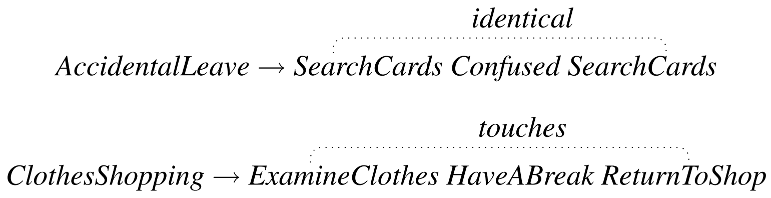

identical
AccidentalLeave → SearchCards Confused SearchCards
touches
ClothesShopping → ExamineClothes HaveABreak ReturnToShop

Figure 5: Two examples for spatial constraints in context-free production rules.

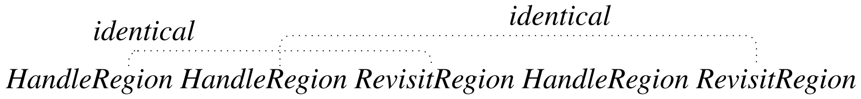

identical
identical
HandleRegion HandleRegion RevisitRegion HandleRegion RevisitRegion

Figure 6: Sequence of intentions with crossing spatial constraints.

We would need the following rule

$$\begin{aligned}(AccidentalLeave, cache_{1,1}) \rightarrow \\ (SearchCards, cache_{1,1}), \\ (Confused, [unconstrained]), \\ (SearchCards, cache_{1,1})\end{aligned}$$

We cannot formulate this in SGIS, but still it makes no sense to write rules for pairs of (intention, region). What we would need to formalize the *accidental leaving pattern* elegantly is displayed in Fig. 5, top. We can easily find other examples of the pattern 'a certain behavior/intention occurs in a region which has a spatial relation r to another region where the agent has done something else before'. For instance, we can find use cases where it makes sense to detect a $ReturnToX$ intention if the agent has forgotten the way back to some place. We could define this as 'the agent shows a searching behavior in a region which $touches$ a region she has been to before', see Fig. 5, bottom.

3.2 Cross-dependencies: a parallel to NLP

Two or more 'return to region' intentions can easily be crossed, see Fig. 6. In a real CityPoker game this can happen for tactical reasons. Players in CityPoker do not necessarily change a playing card although they have found it. They memorize the types of cards they have found and their exact position, and continue in the game. For a number of reasons it might make sense to change in another cache_region first. Sometimes they return to that cache_region at some time later in the game to change a card (without the effort of answering the quiz, cache search, and so on). What we need for this *crossed return to region pattern* is a possibility to create *cross-dependencies*.

3.3 Spatially Constrained TAGs

To express the spatial dependencies described above, we take TAGs as defined in (Joshi and Schabes, 1997) with links as described in (Joshi, 1985), and enhance them by spatial knowledge.

Definition: A Spatially Constrained Tree-Adjoining Grammar is defined as $SCTAG = (TAG, R, SR, GC, NLC)$, where

- *TAG* = (I, B, IT, AT, S), defined over intentions I, and behaviors B.
- *R* is a set of regions
- *SR* is a set of spatial relations, where each relation $r \subseteq R \times R$
- *GC* $\subseteq (IT \cup AT) \times R$ is a set of grounding constraints
- *NLC* is a set of spatial non-local constraints. Each constraint has a type from the spatial relations *SR* and is defined for two nodes in one tree from *IT* ∪ *AT*.

Adjoining and substitution on an SCTAG work as in (Joshi and Schabes, 1997). The grounding constraints allow us to state that an elementary tree may only be located in a certain number of regions. The non-local constraints, on the other hand, allow us to state that the region of one symbol in an elementary tree must have a certain spatial relation to the region of another symbol in the same elementary tree.

As in SGIS, the terminals and non-terminals at the time of writing an SCTAG are not pairs of (symbol, region), but simply behaviors and intentions. This supports the intuition of a knowledge engineer who first writes the decomposition of intentions to sub-intentions, and in a second step annotates spatial knowledge.

Figure 7 lists part of a SCTAG that handles the re-visisting of cache_regions in CityPoker. Non-local spatial constraints are displayed as dotted lines. A complete grammar for this use case would convert all context-free rules from Fig. 3 to trees and add them to the grammar. This step is trivial. Figure 8 demonstrates how cross-dependencies evolve through two adjoining operations.

3.4 Parsing of SCTAG

For parsing a spatially constrained grammar, we modify Joshi's existing Early-like parsing algorithm (Joshi and Schabes, 1997). Like the original Earley parser for CFG, this parser works on charts in which the elementary constructs of the grammar are kept. In Joshi's parser the 'Earley dot' traverses trees and not Strings. In our case, we additionally store for each symbol in each chart entry the set of regions in which it may occur, i.e. when inserting a new chart entry we resolve the spatial constraints by a simple look-up in the spatial relation table.

The parser works in four steps: scan, predict, complete, and adjoin. We modify the scan operation. The scan operation reads the next symbol from the input and matches it with the chart entries. Although we write our SCTAG rules on intentions and behaviors, we get pairs of *(symbol, region)* during parsing. We first execute scan using *symbol*, as in the original parser, and then use the *region* information to throw away those regions in our chart entries that are not consistent with the *region* information. As soon as a symbol in a chart entry has an empty set of possible regions we throw away the chart entry.

Although we do not provide a formal description of the parser in this paper, it should be clear that adding spatial constraints to such a parser will not make it slower but faster. The reason is that spatial constraints give us more predictive information. 'Any algorithm should have enough information to know which tokens are to be expected after a given left context' (Joshi and Schabes, 1997, p.36). Knowing the spatial context of left-hand terminals we can throw away those hypotheses that are not consistent with the spatial constraints. A formal description of the parser, as well as an evaluation, will be issue of future publications.

4 Related Work

Approaches for IR differ in the way possible intentions are represented. A number of formalisms has been proposed for modeling the mental state of an agent, ranging from finite state machines (Dee and Hogg, 2004) to complex cognitive modeling architectures, like the ACT-R architecture (Anderson et al., 2004). With formal grammars, which are between these two extremes, we try to keep the balance between expressiveness and computational complexity. Another important line of research in IR are approaches based on probabilistic networks, e.g. (Bui, 2003; Liao et al., 2007).

For the classification of segments in Fig. 2 we used a simple decision tree. The set of behavior types we are interested in was chosen manually. An automatic detection of motion patterns is the concern of the spatio-temporal data mining community, see e.g. (Laube et al., 2004).

Spatial constraints are also dealt with in multi-model interfaces supporting sketching, like the nuSketch system (Forbus et al., 2001). Speech recognition provides help 'for stating what spatial relationships are essential versus accidental' (p. 5).

5 Outlook

As a next step we will specify the parsing algorithm for SCTAG formally, and implement it for a mobile device. In this paper we treated all spatial relations as arbitrary relations, without using the formal properties of these relations for inference (like transitivity). Adding temporal constraints could also be worthwhile.

References

Anderson, J.R., D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. 2004. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, October.

Baus, J., A. Krueger, and W. Wahlster. 2002. A resource-adaptive mobile navigation system. In

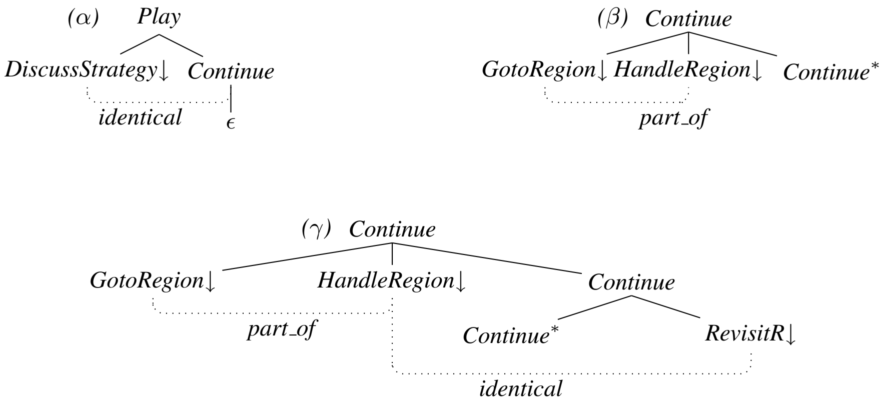


Figure 7: Initial tree (α) and auxiliary trees (β and γ) in a SCTAG for CityPoker.

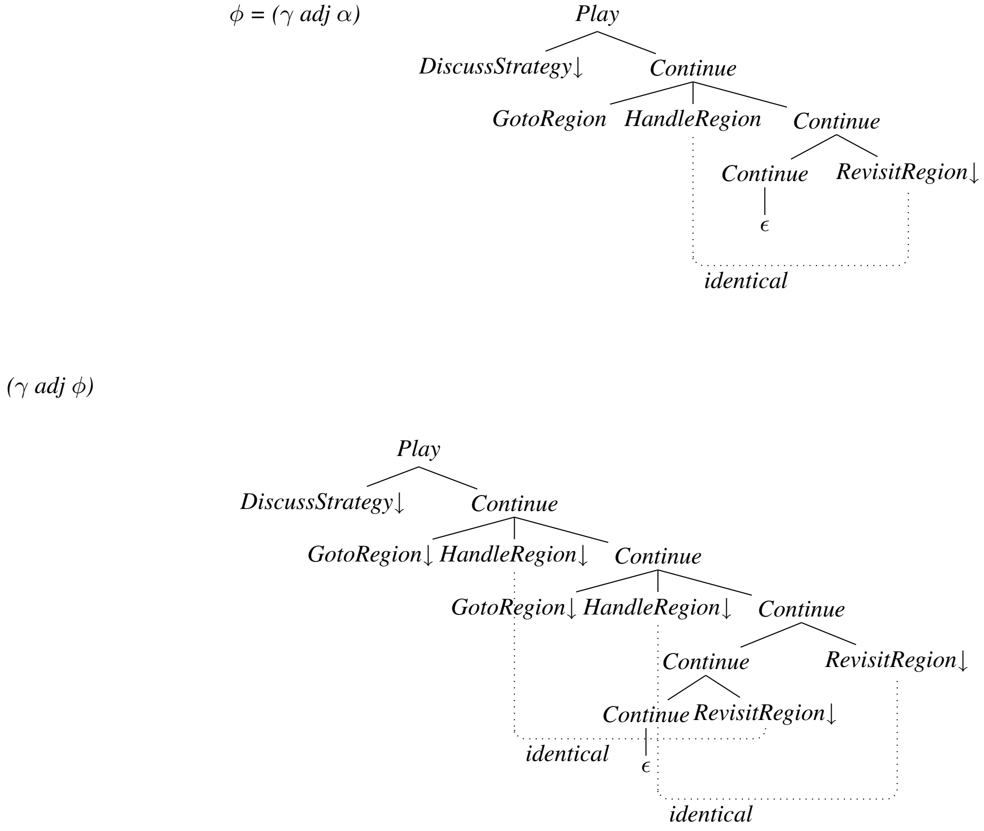


Figure 8: Adjoining in an SCTAG can lead to cross-dependencies of constraints. Non-crossing spatial constraints are omitted for reasons of clarity.

Proc. 7th International Conference on Intelligent User Interfaces, pages 15–22, San Francisco, USA. ACM Press.

Bobick, A.F. and Y.A. Ivanov. 1998. Action recognition using probabilistic parsing. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, pages 196–202.

Bui, Hung H. 2003. A general model for online probabilistic plan recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Carberry, Sandra. 2001. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48.

Chanda, Gaurav and Frank Dellaert. 2004. Grammatical methods in computer vision: An overview. Technical Report GIT-GVU-04-29, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA, November. ftp://ftp.cc.gatech.edu/pub/gvu/tr/2004/04-29.pdf.

Dee, H.M. and D.C. Hogg. 2004. Detecting inexplicable behaviour. In *Proceedings of the British Machine Vision Conference*, pages 477–486. The British Machine Vision Association.

Egenhofer, Max J. and Robert D. Franzosa. 1991. Point-set topological relations. *International Journal of Geographical Information Systems*, 5(2):161–174.

Forbus, Kenneth D., Ronald W. Ferguson, and Jeffery M. Usher. 2001. Towards a computational model of sketching. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 77–83, New York, USA. ACM.

Geib, Christopher W. and Robert P. Goldman. 2003. Recognizing plan/goal abandonment. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1515–1517.

Geib, Christopher W. and Mark Steedman. 2007. On natural language processing and plan recognition. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1612–1617.

Joshi, Aravind K. and Yves Schabes. 1997. Tree-adjoining grammars. In Rozenberg, G. and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–124. Springer, Berlin, New York.

Joshi, A. K. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Dowty, D. R., L. Karttunen, and A. M. Zwicky, editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pages 206–250. Cambridge University Press, Cambridge.

Kautz, Henry and James F. Allen. 1986. Generalized plan recognition. In *Proc. of the AAAI conference 1986*.

Kiefer, Peter and Christoph Schlieder. 2007. Exploring context-sensitivity in spatial intention recognition. In *Workshop on Behavior Monitoring and Interpretation, 40th German Conference on Artificial Intelligence (KI-2007)*, pages 102–116. CEUR Vol-296. ISSN 1613-0073.

Kortuem, G., M. Bauer, and Z. Segall. 1999. Netman: The design of a collaborative wearable computer system. *Mobile Networks and Applications*, 4(1):49–58, March.

Krüger, A., A. Butz, C. Müller, C. Stahl, R. Wasinger, K.-E. Steinberg, and A. Dirschl. 2004. The connected user interface: realizing a personal situated navigation service. In *Proc. of the 9th international Conference on Intelligent User Interfaces*, pages 161–168. ACM Press.

Laube, Patrick, Marc van Krefeld, and Stephan Imfeld. 2004. Finding remo - detecting relative motion patterns in geospatial lifelines. In *Developments in Spatial Data Handling, Proceedings of the 11th International Symposium on Spatial Data Handling*, pages 201–215.

Liao, Lin, Donald J. Patterson, Dieter Fox, and Henry Kautz. 2007. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331.

Pynadath, David V. 1999. *Probabilistic Grammars for Plan Recognition*. Ph.D. thesis, The University of Michigan.

Schlieder, Christoph and Anke Werner. 2003. Interpretation of intentional behavior in spatial partonomies. In *Spatial Cognition III, Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning*, volume 2685 of *Lecture Notes in Computer Science*, pages 401–414. Springer.

Schlieder, C., P. Kiefer, and S. Matyas. 2006. Geogames - designing location-based games from classic board games. *IEEE Intelligent Systems*, 21(5):40–46, Sep/Oct.

Schlieder, Christoph. 2005. Representing the meaning of spatial behavior by spatially grounded intentional systems. In *GeoSpatial Semantics, First International Conference*, volume 3799 of *Lecture Notes in Computer Science*, pages 30–44. Springer.

Stein, Klaus and Christoph Schlieder. 2005. Recognition of intentional behavior in spatial partonomies. In *ECAI 2004 Worskhop 15: Spatial and Temporal Reasoning (16th European Conference on Artificial Intelligence)*.