Realising the Translation of Utterances into Meanings by Petri Net Transducers

Robert Lorenz and Markus Huber

Department of Computer Science
University of Augsburg
robert.lorenz@informatik.uni-augsburg.de

Abstract: In this paper, we illustrate by a small case study how the translation of utterances into meanings within a hierarchical cognitive dynamic speech signal processing system can be realised by Petri net transducers (PNTs).

PNTs are a natural generalisation of finite state transducers (FSTs) for the translation of partial languages consisting of partial words (with a partial order on their symbols) instead of (linear) words (having a total order on their symbols).

For the considered case study we extend previous definitions of PNTs by weights and composition operations. We use bisemirings for the set of weights of a PNT.

1 Introduction

Weighted finite state transducers (FSTs) are classical nondeterministic finite automata in which transitions additionally are equipped with output symbols and weights [1]. An important practical application of such transducers is natural language processing. In this application domain, the weights are used to represent probabilities of transition executions. The behaviour of a transducer is defined by a weighted relation between languages over different alphabets (a transducer defines a weighted translation between two languages). For a uniform definition of the behaviour, the set of weights is equipped with the underlying algebraic structure of a semiring. One important feature of weighted FSTs is the possibility of constructing complex FSTs from simpler ones using composition operations. There are already efficient implementations of such operations in standard libraries [7, 8].

In [5] we introduced a generalisation of FSTs through Petri net transducers (PNTs). PNTs are defined (in a natural way) for the translation between so called partial languages. A *partial language* is a generalisation of (classical) languages, containing so called *partial words* not consisting of a total order on their symbols but of a partial order. In [5] we did not yet consider weights and composition operations on PNTs.

The aim of this paper is to examine the application of PNTs to the translation of recognition results on the syntactic level into semantic interpretations within a hierarchical cognitive dynamic speech signal processing system (as introduced in [2, 9]) through a small case study. Within this system, an acoustic signal is translated over several levels of abstraction into a recognition result on the syntactic level via FSTs. In a next step, recognition results on the syntactic level are translated into semantic interpretations, so called meanings. A common possibility for the representation of meanings are acyclic directed graphs [10, 3]. Since such graphs can be represented by partial orders, FSTs are not longer suitable in this case. Therefore we propose to realise this translation by PNTs. This requires an adequate extension of the definitions from [5] by weights and composition operations.

Considering weights, it turns out that the algebraic structure of semirings (used for FSTs) needs to be extended to bisemirings similar as in [4] in the case of so called weighted branching automata. Concerning composition operations, it is possible to adapt several operations which are central also in the case of FSTs, such as union, product and language composition. Through language composition, FSTs translating an acoustic signal into a recognition result on the syntactic level can be composed with a PNT translating this result from the syntactic level to the semantic level. In this way it is possible to build hierarchical systems consisting of FSTs on some levels and of PNTs on other levels.

The paper is organised as follows: In section 2 we briefly describe the hierarchical cognitive dynamic speech signal processing system the considered case study is based on. In section 3 we describe the case study for the application of PNTs to the translation between syntactic and semantic level of the system. This section includes the necessary extensions of PNTs by weights and composition operations. Finally, in section 4 we give an outlook on future work.

2 A Hierarchical Cognitive Dynamic Speech Signal Processing System

In this section we briefly describe the design of the hierarchical speech signal processing system our case study is based on. This system was proposed in [2, 9, 5] and figure 1 gives an abstract overview of its analysis part (see [9] for a detailed view). The aim is to control a natural language dialogue, where user queries can be freely formulated. The user advises the system to execute certain actions on certain data objects on his behalf. Thus these actions and objects have to be identified by the system. During the dialogue information is collected until the identification is possible.

In the shown approach, the system successively integrates recognition results of user queries (nodes on different levels of abstraction; analysis of an acoustic signal) into an information-state and generates requests concerning missing information (imagine additional nodes on the right where the arrows point downwards; synthesis of an acoustic signal).

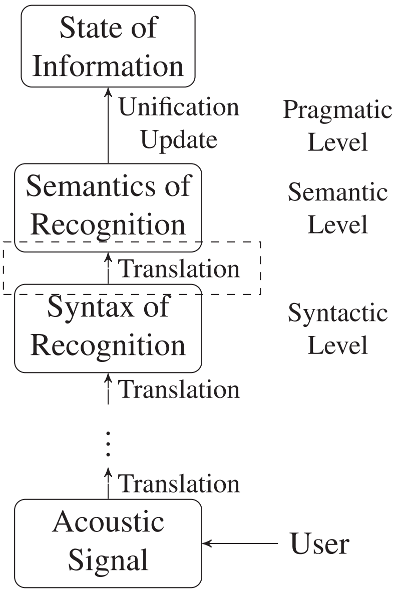

Figure 1 - Analysis side of the hierarchy.

The approach was developed in cooperation with institutes from TU Dresden (R. Hoffmann) and BTU Cottbus (M. Wolff) which are responsible for the lower hierarchical levels, up to the syntactic level. On every level, recognition results are represented by weighted words over appropriate alphabets, where the weights are used to express probabilities. The translation steps between the levels are realised by (weighted) FSTs [2]. Through appropriate composition operations, FSTs are combined for the translation over multiple levels into one single transducer.

The authors of this paper are part of the research team working on the semantic and pragmatic level. The semantic level is used to interpret syntactic recognition results of speech signals. In particular, those results need to be translated from the syntactic level into semantic interpretations (this translation happens within the dotted rectangle of figure 1). Since we use partial orders as modelling language on the semantic and pragmatic level, it is not longer possible to use (classical) FSTs for the translation between those levels.

Within the following case study we propose PNTs for the mentioned translation into semantic interpretations. In [5] we already showed that every FST is a special PNT. In this paper we extend the definitions from [5] by equipping PNTs with weights and introducing the composition of PNTs. This shows, that the whole hierarchy of figure 1 including the semantic level could be completely realised by combining FSTs and PNTs.

3 A Case Study

In this section we provide a small case study for the application of PNTs to the translation of recognition results on the syntactic level into semantic interpretations within the described system. In subsection 3.1 we introduce the world model the case study is based on. A world model describes all semantic interpretations (which we call *meanings*) of utterances (recognised on the syntactic level) the system can deal with. In subsection 3.2 we introduce weighted PNTs and construct a PNT for the translation of utterances into meanings for the previously introduced world model. Such a PNT we call *UMP-Transducer (UMP-T)* (UMP abbreviates Utterance-Meaning-Pair). In subsection 3.3 we introduce the language composition of PNTs and show how the constructed UMP-T can be composed with lower level FSTs.

3.1 World Model

In [3] we introduced a uniform data structure for the representation of all components of the semantic and pragmatic level. This data structure we called *feature-values-relation* (FVR). In particular, we presented a representation of data values together with their semantic interpretations as an FVR. Briefly, an FVR is an acyclic directed graph describing a hierarchy of semantic categories (which we call *features*) which additionally relates data values (which we call *values*) to features and IDs of data objects to values (actions are modelled as features).

Each concrete application of the introduced system is based on a *world model* which is given as an FVR and describes all data objects, values and features the system can cope with. To keep the example simple and to obtain smaller graphics we assume that the only action possible is to call a person. Therefore we leave out any action-part. We consider the (data-object) features *person*, *firstname* and *lastname*, where *firstname* and *lastname* are sub-features of *person*, i.e. *person* is described by (consists of) these two features. The features *firstname* and *lastname* are not related, obviously. Again for simplicity we do not consider other features such as for example *address* and relations between features like *person* “lives at” *address* which also can be modelled using FVRs. For the example it is only possible to describe the person which should be called by its first-name and last-name. From now on we abbreviate the features *person*, *firstname* and *lastname* by *P*, *FN* and *LN*, respectively.

Assume that the world model includes exactly the following three persons given by their names: Parker Lewis, Peter Parker and Pete Rapaka. The world model relates their names to the corresponding features and different object IDs to their first- and last-names as illustrated in figure 2.

The world model not only describes all semantic interpretations but completely determines all utterances which can be recognised. These are all utterances which have a semantic interpretation within the world model, i.e. which can be mapped to a part of the world model. In general, different utterances may have the same semantic interpretation and there may be different possible semantic interpretations of one utterance.

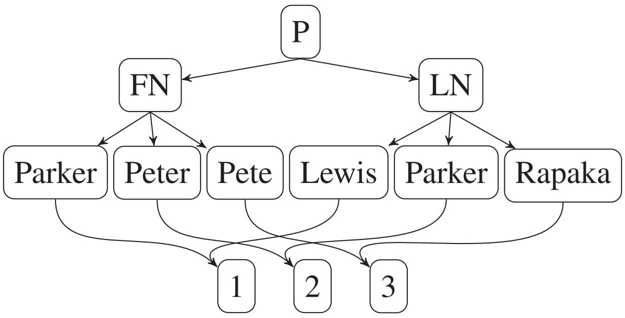

Figure 2 - A simple world model containing three persons.

3.2 UMP-Transducers

In this subsection we first introduce PNTs and then construct a concrete PNT for the translation of utterances into meanings within the world model from the previous subsection. We only give an informal description and use a very basic syntax of Petri nets for those which are not familiar with their theory. For a detailed, complete and formal introduction to PNTs we refer to [6].

A Petri net consists of transitions (drawn as rectangles), places (circles) and a flow relation between places and transitions (directed edges). The flow relation assigns pre- and post-conditions to transitions. The state of a PNT is given by a marking of some places (tokens in places). If a place is marked, then the corresponding condition is satisfied. Figure 3 shows three Petri nets with transition names drawn inside the transitions.

The occurrence of a transition is possible, if all of its pre-conditions are satisfied. Its occurrence leads to a state where none of its pre-conditions and all of its post-conditions are satisfied. The main difference to automata is that the state of a PNT is distributed over several locations. If, in some state, two transitions do not share pre-conditions and all pre-conditions of both transitions are satisfied, then both transitions may occur independently in any order or also simultaneously. Such transitions are called *concurrent (in the considered state)*. This makes it possible to define the occurrence of *step sequences*, where each step is a set of concurrent transitions. For example, in the net N_1 in figure 3 the step sequence $\{a_1\}\{b_1\}\{c_1\}\{d_1\}$ can occur and in the net N_3 from the same figure the step sequences $\{a_3\}\{b_3,e_3\}\{c_3,f_3\}\{d_3\}$ or $\{a_3\}\{b_3\}\{c_3,e_3\}\{f_3\}\{d_3\}$ can occur. More general, it is possible to define *partially ordered runs (po-runs)* of a Petri net. Such a run is a partially ordered set of nodes labelled by transition names, called *LPO*. The nodes (drawn as small filled circles) represent transition occurrences and the partial order (drawn by directed edges) an "earlier than"-relation between them in the sense that one transition occurrence can be observed earlier than another transition occurrence. If there are no arrows between two transition occurrences, then these transition occurrences are concurrent in the above described sense. An LPO is a po-run of a net, if it is consistent with the set of step sequences which can occur in the net. In figures, in general we do not show the names of the nodes of an LPO, but only their transition name labels and we often omit transitive arrows of LPOs for a clearer presentation. Figure 3 shows a po-run for each of the shown nets.

A *Petri net transducer (PNT)* is a Petri net where each transition is augmented with an *input label* and an *output label*. These labels may be symbols from specific alphabets or the empty word symbol ε. For every transition occurrence, a PNT may read a symbol x from an input alphabet Σ and may print a symbol y from an output alphabet Δ. Graphically, these symbols are annotated to transitions in the form $x:y$. If no input symbol should be read or no output symbol should be printed, we use ε as annotation. Each PNT has an initial and a final state, which are both defined by single places. We only consider po-runs, which can occur in the initial state and lead to the final state. The set of all such po-runs of a PNT N we denote by $\mathit{LPO}(N)$.

An *input word* of a PNT is defined as a po-run of the net with nodes relabelled with input symbols (where ε-labelled nodes are deleted). Analogously, the *output word* corresponding to an input word is built through relabelling nodes with output symbols. For LPOs u over Σ and v over Δ, we denote by $\mathit{LPO}(N,u)$ the subset of all LPOs from $\mathit{LPO}(N)$ with input label u, and by $\mathit{LPO}(N,u,v)$ the subset of all LPOs from $\mathit{LPO}(N,u)$ with output label v.

Translating input words into output words, a PNT provides a technique for translation of LPOs over an input alphabet into LPOs over an output alphabet. Figure 3 shows three PNTs with associated po-runs, input words and output words. The PNTs N_1 and N_2 have two different utterances on the syntactic level as input and no output. Such an utterance is a sequence of words and may be represented by a total order. The PNT N_3 has no input and a meaning as output. A meaning is an FVR which is consistent with the world model and defines a possible semantic interpretation of an utterance by relating values occurring in the utterance to features and IDs of data objects. Since we can identify an FVR with its transitive closure, meanings can be viewed as partial orders.

Observe that within the considered world model the inputs of N_1 and N_2 define two alternative

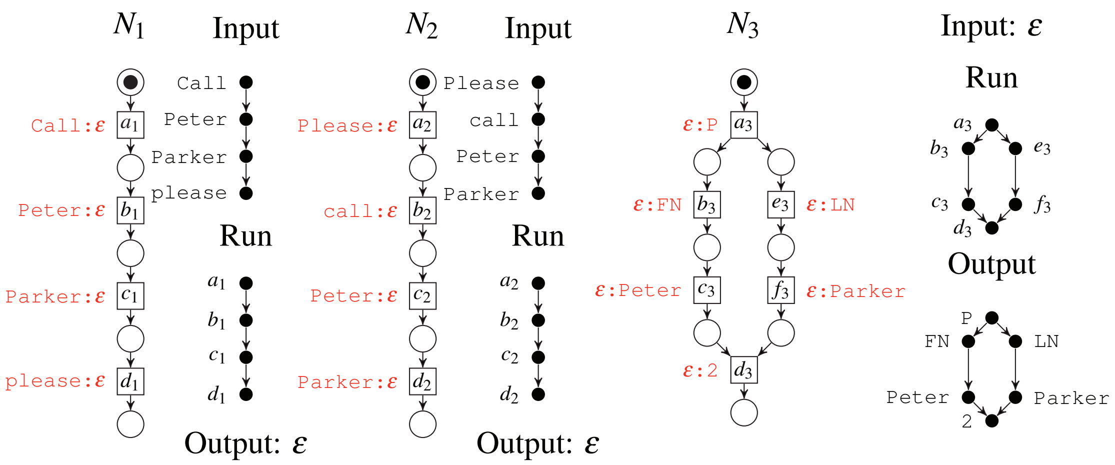

Figure 3 - Three PNTs with po-runs and associated input words and output words.

utterances having the (same) meaning given by the output of N_3, i.e. both relate to the first-name and last-name of the person with ID 2. It is possible to define a PNT N_4 translating these utterances into their meaning using the composition operations $\oplus$ for the *union* and $\otimes$ for the *product* of PNTs through $N_4 = (N_1 \oplus N_2) \otimes N_3$.

Before we formally define these (and other) composition operations, we need to introduce weights of PNTs in order to reflect probabilities of recognition results. Weighted PNTs are PNTs in which each transition additionally carries some weight. Graphically, a weight $\omega(t)$ is annotated to a transition t in the form $/\omega(t)$. The weights are elements of an algebraic structure called bisemiring. A *bisemiring* is a six-tuple $\mathscr{S} = (S, \oplus, \otimes, \boxtimes, \overline{0}, \overline{1})$, where $\oplus$, $\otimes$ and $\boxtimes$ are binary operations on the set S (*S-addition*, *S-sequential multiplication* and *S-parallel multiplication*) satisfying the following assumptions: $\oplus$ is commutative and associative, $\otimes$ is associative and distributing over $\oplus$, $\boxtimes$ is associative and commutative and distributing over $\oplus$, the zero $\overline{0} \in S$ is neutral w.r.t. $\oplus$ and absorbing w.r.t. $\otimes$ and $\boxtimes$, and the unit $\overline{1} \in S$ is neutral w.r.t. $\otimes$. For example, $([0,1], \max, \cdot, \min, 0, 1)$ is a bisemiring, which we will use in all following concrete examples. The $\otimes$-operation is used to compute the weight along paths within a po-run by sequentially multiplying the weights of the transitions. The $\boxtimes$-operation is used to compute the weight of concurrent paths (of transition occurrences) within a po-run by parallel multiplying the weights of these paths. The $\oplus$-operation is used to compute the weight of a pair of input and output words (u, v) by summing up the weights of all po-runs with corresponding input word u and output word v. Figure 4 shows the PNT N_4 together with example weights and with its two po-runs. One po-run defines the input word of N_1 and the output word of N_3 and the other po-run defines the input word of N_2 and the output word of N_3.

In order to define the weight of a po-run, consider a po-run as the synchronous product of all of its lines with maximal length, where a *line* of an LPO is a totally ordered sub-LPO. The set of all maximal lines of an LPO *lpo* we denote by *lines*(*lpo*). For example, the po-run of the net N_3 in figure 3 has the maximal lines $a_3b_3c_3d_3$ and $a_3e_3f_3d_3$. The weight of a line is computed by sequentially multiplying the weights of the transitions, i.e. $\omega(a_3b_3c_3d_3) = \omega(a_3) \otimes \omega(b_3) \otimes \omega(c_3) \otimes \omega(d_3)$. If $\otimes$ is distributive over $\boxtimes$, we define the weight of a po-run *lpo* by $\omega(lpo) = \boxtimes_{lpo' \in lines(lpo)} \omega(lpo')$.

The relation between $\otimes$ and $\boxtimes$ is needed to derive effective constructions for composition operations. The weight is defined in such a way that only the weights of dependent parts of a po-run are sequentially multiplied and the weights of independent parts are parallel multiplied. The bisemiring $([0,1], \max, \cdot, \min, 0, 1)$ satisfies the above condition. For example the left hand side po-run of N_4 has the weight $\min(\omega(abcdefgh), \omega(abcdeijh)) = 0.36$ and the right hand side po-run has the weight $\min(\omega(klmnefgh), \omega(klmneijh)) = 0.54$.

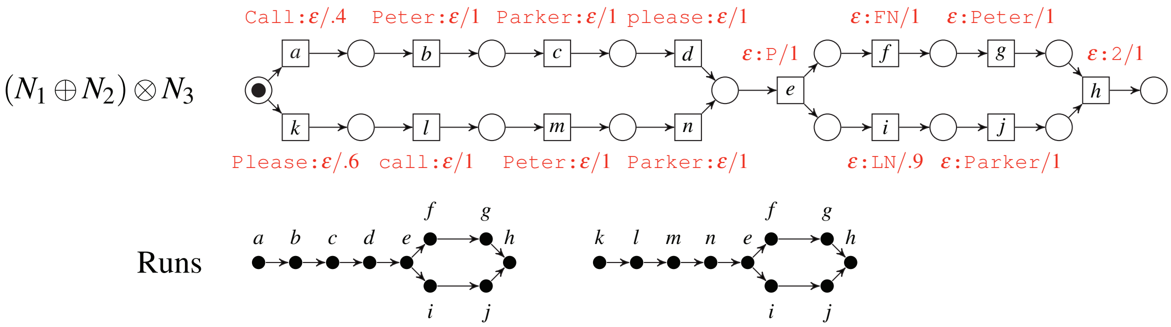

Figure 4 - A weighted PNT together with a po-run.

The *output weight* a PNT assigns to all pairs of LPOs u over Σ and v over Δ is defined through

$$N(u,v) = \bigoplus_{lpo \in LPO(N,u,v)} \omega(lpo),$$

if this sum exists, is an element of the bisemiring and is well-defined (note that the sum may be infinite). If this is the case for all such pairs of LPOs (u,v), the PNT is called *regulated*. For $LPO(N,u,v) = \emptyset$ we set $N(u,v) = \bar{0}$. The *weight* a PNT assigns to v over Δ is computed from its output weights through $N(v) = \oplus_u N(u,v)$. If u_1 denotes the input word of N_1, u_2 the input word of N_2 and v the output word of N_3, then $N_4(u_1,v) = 0.36$, $N_4(u_2,v) = 0.54$ and $N_4(v) = 0.54$.

It is possible to introduce several useful composition operations on regulated PNTs. In general, a composition operation is given in a functional form defining the output weight of the composed PNT based on the output weights of the original PNTs and bisemiring-operations. In a next step it is necessary to find an effective construction, showing that there is a composed PNT having the intended output weight. For example, the *sum (or union)* $N_1 \oplus N_2$ of two PNTs N_1 and N_2 over the same bisemiring, input alphabet Σ and output alphabet Δ is defined as a PNT with the output weight $(N_1 \oplus N_2)(u,v) = N_1(u,v) \oplus N_2(u,v)$.

The *product (concatenation)* $N_1 \otimes N_2$ of two PNTs N_1 and N_2 over the same bisemiring, input alphabet Σ and output alphabet Δ is defined as a PNT with the output weight

$$(N_1 \otimes N_2)(u,v) = \bigoplus_{u=u_1;u_2, v=v_1;v_2} N_1(u_1,v_1) \otimes N_2(u_2,v_2).$$

The sum runs over all possible ways of decomposing an LPO u into a prefix u_1 and a suffix u_2 of the form $u = u_1;u_2$, and similar for v. For both, union and product, there are effective simple constructions of a composed PNT as illustrated by PNT N_4 from figure 4. Note that there are several possibilities for reducing the size of N_4, but it is a topic of future research to develop a general theory for the minimisation and optimisation of PNTs. Other composition operations which can be defined are closure, language composition, parallel product and synchronous product [6]. The operations of union, product, closure and language composition are also central operations in the case of FSTs. The operations of parallel product and synchronous product are new and cannot be applied to FSTs.

3.3 Adding the Semantic Level to the System

Assume the user says “Call Parker, please.” and the speech recogniser assigns the probability .8 to the utterance *Call→Parker→please* and the remaining mass of .2 to other utterances. For the example we restrict ourselves to the one alternative *Call→Rapaka→please* which sounds somewhat similar. Note that the higher weighted utterance may relate to a first-name or a last-name of a person since both cases are covered by the world model. However, on the syntactic

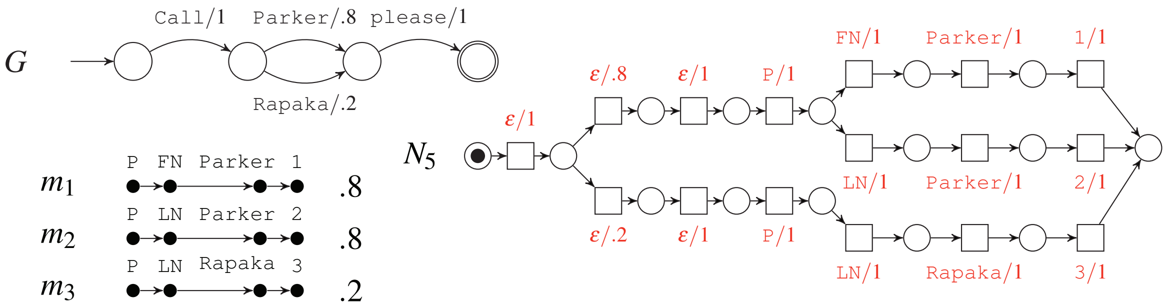

Figure 5 - “Call Parker, please.” gets translated into different weighted meanings m_1, m_2 and m_3.

level a speech recogniser is not able to make a distinction because there is no evidence within the acoustic signal. The recognition result can be represented by the FST G in the upper left part of figure 5. Note that we left out the input symbols and show only a generator instead of a transducer because the input – the acoustic signal – does not matter for the following thoughts.

Two transducers can be language composed if the output alphabet of the first one equals the input alphabet of the second one. This operation can be seen as a chained translation from the input alphabet of the first transducer to the output alphabet of the second one. The language composition $T_1 \circ T_2$ of FSTs T_1 and T_2 is functionally defined via

$$(T_1 \circ T_2)(u,w) = \bigoplus_{v} T_1(u,v) \otimes T_2(v,w),$$

where the sum runs over all $v \in \Delta_1^*$ representing the output label of a path of T_1 and the input label of a path of T_2. In the construction of $T_1 \circ T_2$, a transition t_1 from T_1 is merged with a transition t_2 from T_2 if the output symbol of t_1 coincides with the input symbol of t_2. The weight of the merged transitions is derived by sequential multiplication. The transitions of T_1 having empty output symbol, as well as all transitions of T_2 having empty input symbol are put into an arbitrary but fixed sequence, i.e. weights are sequentially multiplied. This way, the constructed FST has the intended weight.

For PNTs, such a construction does not make sense, since a PNT is able to reflect concurrency. If there is a transition of the first PNT with empty output label, no symbol is printed if it fires. Therefore it should be independent from each transition of the second PNT. A similar argumentation holds for transitions of the second PNT with empty input symbol. The *language composition* $N = N_1 \circ N_2$ of PNTs N_1 and N_2 is constructed by merging transitions in the same situation as for FSTs. The other transitions are reused with unchanged input and output symbols, weights and edges and remain unordered. A functional definition of the composed PNT’s weight is unclear since the concurrency relations between transition occurrences may be complicated.

Since each FST can easily be represented by a PNT [5, 6] we can apply language composition to G and some UMP-T translating the outputs of G into the meanings m_1, m_2 and m_3 shown in figure 5 (this UMP-T can be constructed in a similar way as shown in the last subsection for other utterances and meanings). The result of the composition might look like N_5 on the right of figure 5. Note that we assumed in the figure that all transitions of the UMP-T carried the weight 1. Therefore, the probabilities of the recognition result are promoted to the meanings. The two upper meanings have the same weight because they originated from the same (ambiguous) utterance. N_5 is also only a generator since G is one. Note that N_5 is the result of the language composition over all levels from figure 1 for the acoustic input “Call Parker, please.”. Thus it represents all possible semantic interpretations of the user’s input.

In general, the transition weights of a UMP-T are not equal to $\overline{1}$, but are adjusted during a dialogue. For example, since the system does not know which person to call, it generates a request like “Should I call Parker Lewis or Peter Parker?”. Now it is more likely that the user gives an answer where both a first-name and a last-name are included. Accordingly the weights inside the UMP-T can be adjusted to reflect this expectation. Another possibility for the adjustment of weights is to take the user's preferences into account. If a user often describes a person only by its first-name than this translation should be more likely. In the example the uppermost meaning would then gain a higher weight than the second one although both still originate from the same utterance.

4 Outlook

There are important further steps in several directions. We aim to develop a complete theory of composition and optimisation operations of PNTs including efficient algorithms. For application in semantic dialogue modelling and speech recognition, we also need to find efficient algorithms computing the *N* best po-runs of a PNT. At the time of writing we examine semi-automatic procedures to construct a UMP-T from experimental audio data (generated in Wizard-of-Oz experiments). Moreover, we want to apply the same theory to the synthesis part of the described hierarchical system.

References

[1] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. Monographs in Theoretical Computer Science. Springer, 2009.

[2] R. Hoffmann, M. Eichner, and M. Wolff. Analysis of verbal and nonverbal acoustic signals with the Dresden UASR system. In *Verbal and Nonverbal Communication Behaviours*, volume 4775 of *LNAI*, pages 200– 218. Springer, 2007.

[3] M. Huber, C. Kölbl, R. Lorenz, R. Römer, and G. Wirsching. Semantische Dialogmodellierung mit gewichteten Merkmal-Werte-Relationen. In *Proceedings of "Elektronische Sprachsignalverarbeitung (ESSV)"*, volume 53 of *Studientexte zur Sprachkommunikation*, pages 25–32, 2009.

[4] D. Kuske and I. Meinecke. Branching automata with costs - a way of reflecting parallelism in costs. *Theoretical Computer Science*, 328:53 – 75, 2004.

[5] R. Lorenz and M. Huber. Petri net transducers in semantic dialogue modelling. In *Proceedings of "Elektronische Sprachsignalverarbeitung (ESSV)"*, volume 64 of *Studientexte zur Sprachkommunikation*, pages 286–297, 2012.

[6] R. Lorenz and M. Huber. Towards a theory of weighted petri net transducers. In *Applications and Theory of Petri Nets, 34th International Conference, Proceedings*. submitted, 2013.

[7] M. Mohri. *Weighted Automata Algorithms*, pages 213 – 254. In Droste et al. [1], 2009.

[8] M. Wolff. Akustische Mustererkennung. Habilitation, 2009.

[9] M. Wolff, R. Römer, and R. Hoffmann. Hierarchische kognitive dynamische Systeme zur Sprach- und Signalverarbeitung. In *Proceedings of "Elektronische Sprachsignalverarbeitung (ESSV)"*, volume 64 of *Studientexte zur Sprachkommunikation*, pages 96–103, 2012.

[10] S. Young. Still talking to machines (cognitively speaking). In T. Kobayashi, K. Hirose, and S. Nakamura, editors, *INTERSPEECH*, pages 1–10. ISCA, 2010.