High-Level Nets with Nets and Rules as Tokens

Kathrin Hoffmann[1], Hartmut Ehrig[1], and Till Mossakowski[2]

[1] Institute for Software Technology and Theoretical Computer Science, Technical University Berlin
[2] BISS, Department of Computer Science, University of Bremen

Abstract. High-Level net models following the paradigm "nets as tokens" have been studied already in the literature with several interesting applications. In this paper we propose the new paradigm "nets and rules as tokens", where in addition to nets as tokens also rules as tokens are considered. The rules can be used to change the net structure. This leads to the new concept of high-level net and rule systems, which allows to integrate the token game with rule-based transformations of P/T-systems. The new concept is based on algebraic high-level nets and on the main ideas of graph transformation systems. We introduce the new concept with the case study "House of Philosophers", a dynamic extension of the well-known dining philosophers. In the main part we present a basic theory for rule-based transformations of P/T-systems and for high-level nets with nets and rules as tokens leading to the concept of high-level net and rule systems.

Keywords: High-level net models, algebraic high-level nets, nets and rules as tokens, integration of net theory and graph transformations, case study: House of Philosophers, algebraic specifications, graph grammars and Petri net transformations.

1 Introduction

The paradigm "nets as tokens" has been introduced by Valk in order to allow nets as tokens, called object nets, within a net, called a system net (see [Val98, Val01]). This paradigm has been very useful to model interesting applications in the area of workflow, agent-oriented approaches or open system networks. Especially his concept of elementary object systems [Val01] has been used to model the case study of the hurried philosophers proposed in [Sil01]. In elementary object systems object nets can move through a system net and interact with both the system net and with other object nets. This allows to change the marking of the object net, but not their net structure. According to the requirements of the hurried philosophers in [Sil01] the philosophers have the capability to introduce a new guest at the table, which - in the case of low level Petri nets - certainly changes the net structure of the token net representing the philosophers at the table. We use the notion of token net instead of object net in order to avoid confusion with features of object-oriented modeling. Instead our intention

G. Ciardo and P. Darondeau (Eds.): ICATPN 2005, LNCS 3536, pp. 268–288, 2005.

is to consider the change of the net structure as rule-based transformation of Petri nets in the sense of graph transformation systems [Ehr79, Roz97]. In order to integrate the token game of Petri nets with rule-based transformations, we propose in this paper the new paradigm "nets and rules as tokens" leading to the concept of high-level net and rule systems.

In Section 2 we show how this new concept can be used to model the main requirements of the hurried philosophers [Sil01]. Of course, this concept has interesting applications in all areas where dynamic changes of the net structure have to be considered while the system is still running. This applies especially to flexible workflow systems (see [Aal02]) and medical information systems (see [Hof00]).

In Section 3 we introduce the basic theory for rule-based transformations of P/T-systems. This theory is inspired by graph transformation systems [Ehr79, Roz97], which have been generalized already to net transformations systems in [EHK91, EP04], including high-level and low-level nets. The theory in these papers is based on pushouts in the corresponding categories according to the double-pushout approach of graph transformations in [Ehr79]. In order to improve the intuition of our concepts for the Petri net community we give in this paper an explicit approach of rule-based transformations for P/T-systems, which is new and extends the theory of P/T-net transformations taking into account also initial markings, and avoids categorical terminology like pushouts. Moreover, the interaction of the token game and transformation of nets - as considered in this paper - has not been studied up to now.

In Section 4 we introduce high-level nets with nets and rules as tokens leading to our new concept of high-level net and rule (HLNR) systems motivated above. This new concept is based on algebraic high-level (AHL) nets [PER95] using the terminology of [EHP02]. In order to model nets and rules as tokens we present a specific signature together with a corresponding algebra with specific sorts for P/T-systems and rules. Moreover, there are operations corresponding to firing of a transition and applying a rule to a P/T-system respectively. Since AHL-nets are based on classical algebraic specifications (see [EM85]) we are able to give a set theoretic definition of domains and operations. In order to obtain also an algebraic specification we need algebraic higher-order specifications as presented in HasCasl [Hets, SM02], which allows to specify function types with set-theoretic notions of semantics using intensional algebras.

In Section 5 we discuss specification and implementation aspects for our approach. More precisely, we discuss how the concept of algebraic higher-order (AHO) nets based on HasCasl, which has been already introduced in [HM03], can be used to specify the algebra of HLNR-systems. Since tools for HasCasl already have been implemented [Mos05, Hets] this is an important step towards implementation and tool support for HLNR-systems. Unfortunately, this is not possible using CPN tools [RWL03] for Coloured Petri (CP) Nets [Jen92]. Actually, CP-Nets are based on an extension of the functional language Standard ML [MTH97]. As Standard ML does not allow functional equivalence testing, it is not suitable for our purpose where we need a form of functional equivalence. The conclusion in Section 6 includes proposals for future work.

2 Case Study: House of Philosophers

In order to illustrate the concepts described in Section 3 and Section 4 we will present a small system inspired by the case study "the Hurried Philosophers" of C. Sibertin-Blanc proposed in [Sil01] which is a refinement of the well-known classical "Dining Philosophers".

Requirements. In our case study "House of Philosophers" presented below we essentially consider the following requirements:

1. There are three different locations in the house where the philosophers can stay: the library, the entrance-hall, and the restaurant;
2. In the restaurant there are different tables where one or more philosophers can be placed to have dinner;
3. Each philosopher can eat at a table only when he has both forks, i.e. the philosophers at each table follow the rules of the classical "Dining Philosophers";
4. The philosophers in the entrance-hall have the following additional capabilities:
   (a) They are able to invite another philosopher in the entrance-hall to enter the restaurant and to take place at one of the tables;
   (b) They are able to ask a philosopher at one of the tables with at least two philosophers to leave the table and to enter the entrance-hall.

System Level. In Fig. 1 we present the system level of our version of the case study. The system level is given by a high-level net and rule system, short HLNR-system, which will be explained in Section 4. The marking of the HLNR-system shows the distribution of the philosophers at different places in the house and the firing behavior of the HLNR-system describes the mobility of the philosophers. There are three different locations in the house where the philosophers can stay: the library, the entrance-hall, and the restaurant. Each location is represented by its own place in the HLNR-system in Fig. 1. Initially there are two philosophers at the library, one philosopher at the entrance-hall, and four additional philosophers are at table 1 resp. table 2 (see Fig. 5 and Fig. 6) in the restaurant.

Philosophers may move around, which means they might leave and enter the library and they might leave and enter the tables in the restaurant. The mobility aspect of the philosophers is modeled by transitions termed *enter* and *leave library* as well as *enter* and *leave restaurant* in our HLNR-system in Fig. 1. While the philosophers are moving around, the static structure of the philosophers is changed by rule-based transformations. E.g. a philosopher enters the restaurant and arrives at a table. Then the structure and the seating arrangement of the philosophers have to be changed. For this reason, we have tokens of type *Rules*, $rule_1$, ..., $rule_4$, which are used as resources. Because the philosophers have their own internal behavior, there are two transitions, *start/stop reading* and *start/stop activities*, to realize the change of the behavior.

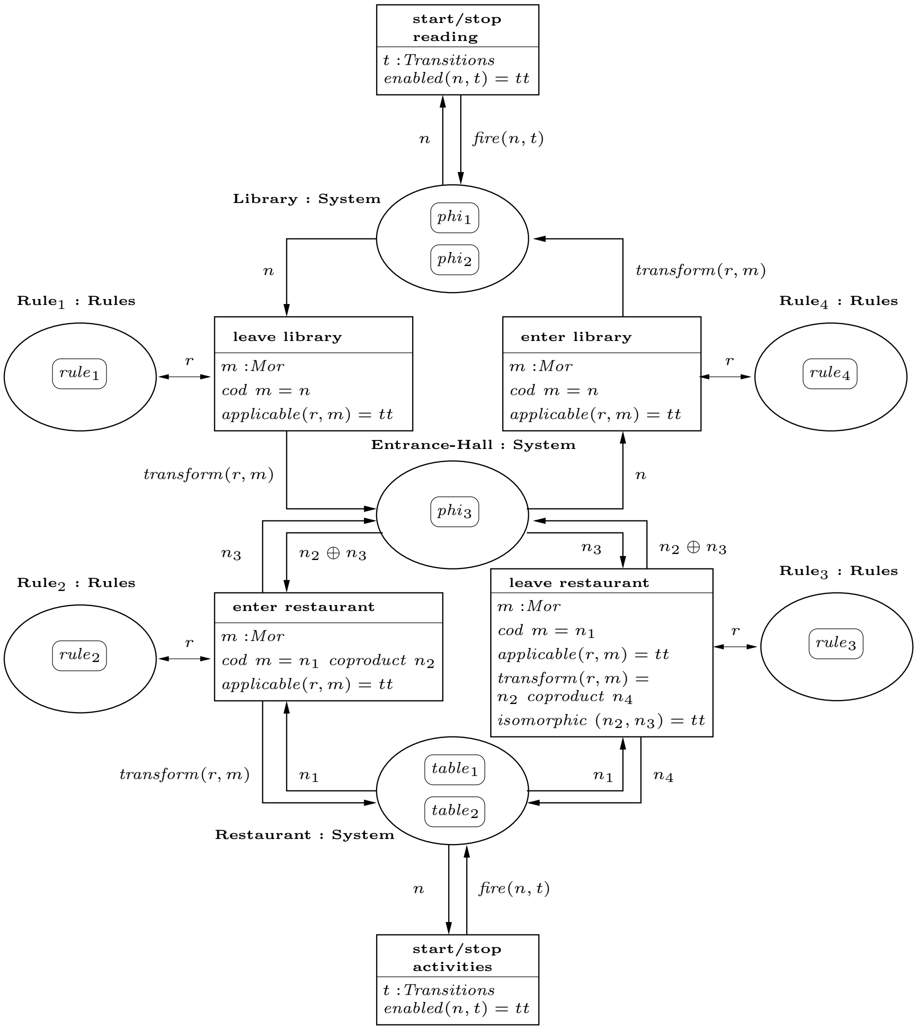


Fig. 1. High-level net and rule system of "House of Philosophers"

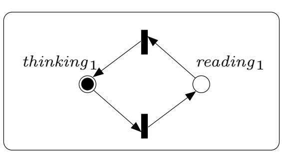


Fig. 2. Token net phi_1 of philosopher 1

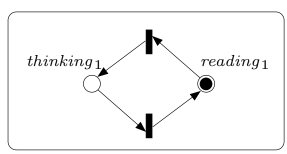


Fig. 3. Token net phi'_1 of philosopher 1

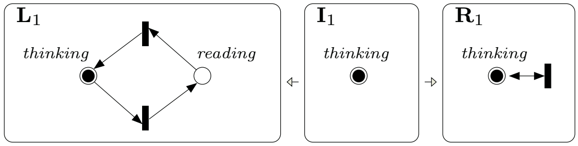


Fig. 4. Token rule of rule $rule_1$

Token Level. The token level consists of two different types of tokens: P/T-systems and rules. They are represented as tokens in the places typed *System* and *Rules* of the HLNR-system in Fig. 1. The tokens on system places are modeled by P/T-systems, i.e. Petri nets with an initial marking. In Fig. 2 the net phi_1 of the philosopher 1 is depicted, which - in the state *thinking* - is used as a token on the place *Library* in Fig. 1. To start reading, we use the transition *start/stop reading* of the HLNR-system in Fig. 1. First the variable n is assigned to the net phi_1 of the philosopher 1 and the variable t to a transition $t_0 \in T_0$ where T_0 is a given vocabulary of transitions. The condition *enabled(n,t)=tt* means that under this assignment t_0 is an enabled transition in the net of phi_1. The evaluation of the term $fire(n, t)$ computes the follower marking of the net (i.e. token $reading_1$) and we obtain the new net phi'_1 of the philosopher 1 depicted in Fig. 3.

Mobility of Philosophers by Application of Rules. We assume that the philosopher 1 wants to leave the library, i.e. the transition *leave library* in the HLNR-system in Fig. 1 must fire. For this purpose we have to give an assignment for the variables n, r and m in the net inscriptions of the transition. They are assigned to the net phi_1 (see Fig. 2), the rule $rule_1$ (see Fig. 4), and a match morphism $m_1 : L' \to G$ between P/T-systems. The first condition *cod m=n* requires $G = phi_1$ and the second condition *applicable(r,m)=tt* makes sure that rule $rule_1$ is applicable to phi_1, especially $L' = L_1$, s.t. the evaluation of the term *transform(r,m)* leads to the new net phi''_1 isomorphic to R_1 of $rule_1$ in Fig. 4. As result of this firing step phi_1 is removed from place *Library* and phi''_1 is added on place *Entrance-Hall*. In general, a rule $r = (L \xleftarrow{i_1} I \xrightarrow{i_2} R)$ is given by three P/T- systems called left-hand side, interface, and right-hand side respectively.

In a further step the philosopher 1 is invited by the philosopher 3 to enter the restaurant in order to take place as a new guest at the table 1. The philosopher 3 accompanies philosopher 1 but returns to the entrance-hall. The token net phi_3 of philosopher 3 is isomorphic to R_1 of $rule_1$ in Fig. 4 where *thinking* in R_1 is replaced by $thinking_3$. Currently the philosophers 4 and 5 are at the table 1 (see Fig. 5). Both philosophers may start eating, but apparently compete for their shared forks, where *left fork*$_4$=*right fork*$_5$ and *left fork*$_5$=*right fork*$_4$. Analogously table 2 has the same net structure as table 1 but different philosophers are sitting at table 2 (see Fig. 6). To introduce the philosopher 1 at the table 1 the seating arrangement at table 1 has to be changed. In our case the new guest takes place between philosopher 4 and 5. Formally, we apply rule $rule_2 = (L_2 \xleftarrow{i_1} I_2 \xrightarrow{i_2} R_2)$, which is depicted in the upper row of Fig. 7 and used as token on place $Rule_2$. We have to give an assignment v for the variables

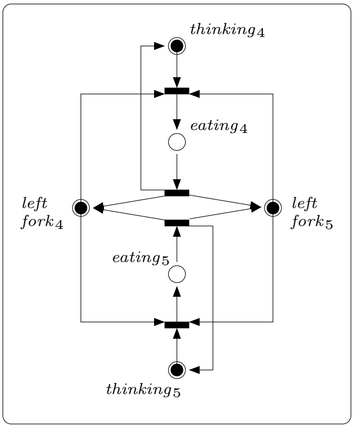


Fig. 5. Token net $table_1$ of philosopher 4 and 5 at table 1

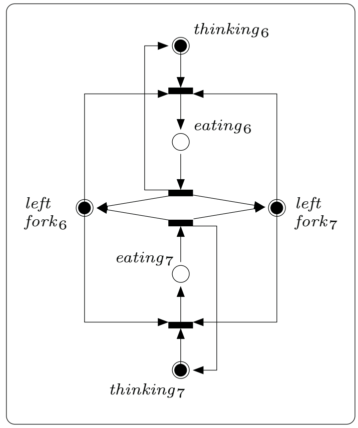


Fig. 6. Token net $table_2$ of philosopher 6 and 7 at table 2

of the transition *enter restaurant*, i.e. variables n_1, n_2, n_3, r, and m. The assignment v is defined by $v(n_1) = table_1$, $v(n_2) = phi_1''$, $v(n_3) = phi_3$, $v(r) = rule_2$, and $v(m) = g$ (see match morphism $g : L_2 \to G$ in Fig. 7). Then we compute the disjoint union of the P/T-system phi_1'' and the P/T-system $table_1$ as denoted by the net inscription n_1 *coproduct* n_2 in the firing condition of the transition *enter restaurant*. The result is the disjoint union of both nets shown as P/T-system G in Fig. 7.

In our case the match g maps $thinking_j$ and $eating_j$ in L_2 to $thinking_4$ and $eating_4$ in G of Fig. 7. The condition $cod\ m = n_1$ *coproduct* n_2 makes sure that the codomain of g is equal to G. The second condition *applicable(r,m)=tt* checks if $rule_2$ is applicable with match g to G (see "gluing condition" (Def. 4) and "applicability" (Def. 5) in Section 3). In the direct transformation shown in Fig. 7 we delete in a first step $g(L_2 \setminus I_2)$ from G leading to P/T-system C. Note, that a positive check of the "gluing condition' makes sure that C is a well-defined P/T-system (see Prop. 2 in Section 3). In a second step we glue together the P/T-systems C and R_2 along I_2 leading to P/T-system H in Fig. 7. H shows the new version of table 1 given by the net $table_1'$ of table 1, where philosophers 1, 4, and 5 are sitting at the table, all of them in state *thinking*. The effect of firing the transition *enter restaurant* in Fig. 1 with assignments of variables as discussed above is the removal of P/T-systems phi_1'' from place *Entrance Hall* and $table_1$ from place *Restaurant* and adding P/T-System $table_1'$ to the place *Restaurant*.

Philosophers in the entrance-hall have the capability to ask one of the philosophers in the restaurant to leave; this is realized in our system by the transition *leave restaurant* in Fig. 1. We use the rule $rule_3$ defined as inverse of $rule_2$ in Fig. 7, i.e. $rule_3 = (R_2 \xleftarrow{i_2} I_2 \xrightarrow{i_1} L_2)$, which is present as a token on place $Rule_3$. This rule is applied with inverse direct transformation to that depicted in Fig. 7. Finally, the rule $rule_4$ is the inverse of rule $rule_1$ (see Fig. 4), enabling

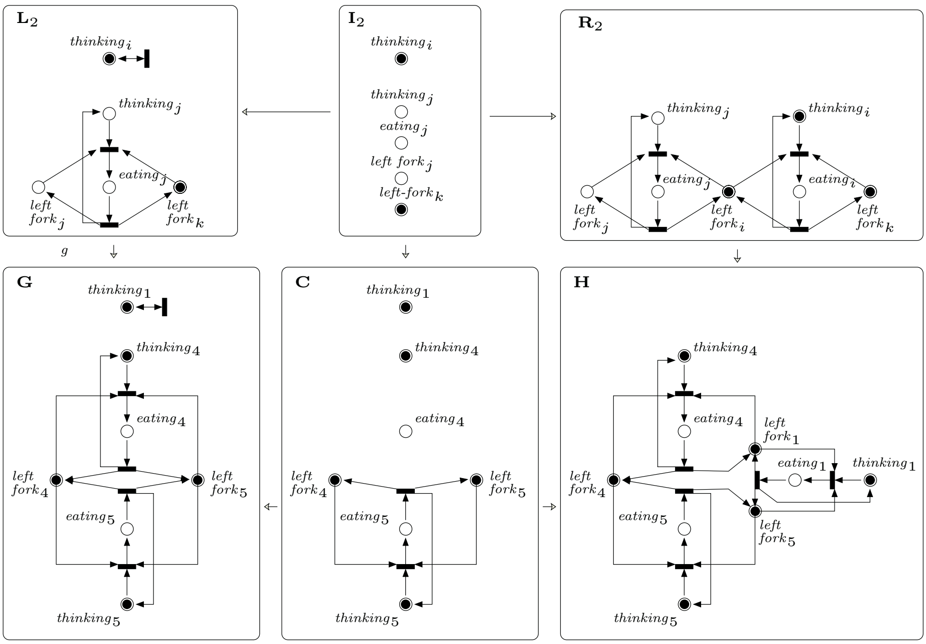


Fig. 7. Direct Transformation

the philosopher to enter the library by firing of the transition *enter library* in Fig. 1. We have to guarantee that after the application of $rule_3$ the philosopher who is leaving the restaurant goes into the entrance-hall. In our case one philosopher is asked by philosopher 3 in the entrance-hall to leave the table. Formally this is denoted by the firing condition $isomorphic(n_2, n_3) = tt$ which ensures that the net of the philosophers denoted by n_2 is isomorphic to the net phi_3 of philosopher 3 denoted by n_3.

The execution of philosopher activities at different tables, i.e. the firing of the transition *start/stop activities* in Fig. 1, is analogously defined as the firing of the transition *start/stop reading* described above.

Validation of Requirements. Our case study "House of Philosophers" satisfies the requirements presented in the beginning of this section.

1. The three different locations in the house are represented by places *Library*, *Entrance-Hall*, and *Restaurant* in Fig. 1;
2. In the initial state we have the two tables $table_1$ with philosophers 4 and 5 and $table_2$ with philosophers 6 and 7 on place *Restaurant*. In a later state also philosopher 1 is sitting at $table_1$ as shown by net H of Fig. 7;
3. If there are $n \geq 2$ philosophers sitting at each table, the table with n philosophers is presented by the classical "Dining Philosophers" net;
4. The capability of a philosopher in the entrance-hall to invite another philosopher to enter (leave) the restaurant is given by firing of the transition

enter restaurant (leave restaurant) in Fig. 1. The applicability of the rule $rule_3$ ensures that a philosopher only leaves a table with at least two philossphers.

Related Work. In [ADC01] there are several other solutions for the case study "the Hurried Philosophers" modeled by different kinds of (high-level) net classes. Most of these approaches integrate object-oriented modeling and Petri nets, including e.g. inheritance, encapsulation, and dynamic binding, etc. In this paper we do not need features of object-oriented modeling. But it is an interesting aspect to extend our approach by integration of these features.

In the solution of the case study using elementary object systems [Val01], each philosopher has his own place and the exchange of forks between the philosophers is realized by an interaction relation. By contrast in our case each table is modeled by its own P/T-system and describes the states and the seating arrangement of present philosophers. Moreover we use rule-based transformations to change the structure of P/T-systems, especially the states and the seating arrangement. In the sense of object-oriented modeling it might be considered to split up the table with philosophers into a net table with only the table properties and nets for each philosopher at the table. In fact our approach allows to model such self-contained components but this would lead to a much more complex model. The advantage of our approach compared with elementary object systems is a more flexible modeling technique. While the HLNR-system in Fig. 1 is fixed we can add further philosophers and philosophers at tables by adding further tokens of type *System* to our model. Analogously we can add further token rules to realize other kinds of transformations.

Note, that elementary object systems [Val01] allow a simple notion of nets as tokens, such that most principles of elementary net theory are respected and extended. Here on the one hand the system-object interaction relation consists of transitions in the system net and transitions in the object net which have to be fired in parallel, and on the other hand the object-object interaction relation guards the parallel firing of transitions in different object nets. By contrast, we are using different formal frameworks for the token level and the system level. In order to integrate interaction relations into our concept of HLNR-system we can extend the signature and the algebra of the algebraic high-level net by appropriate operations and formulate the dependencies between transitions in the firing conditions of the HLNR-system. In this way we can show that elementary object systems can be translated into semantically equivalent HLNR-systems extended by interaction relations.

The idea of controlled modification of token nets is discussed in the context of linear logic Petri nets [Far99] and feature structure nets [Wie01]. The difference to our approach is that in those approaches, the modification is not carried out by rule tokens, but by transition guards. We are not restricted to define a specific token rule for each transition, but we are able to give a (multi-)set of token rules as resources bound to each transition, which realize the local replacement of subnets.

3 Rule-Based Transformation of P/T-Systems

In this section we present rule-based transformations of nets following the double-pushout (DPO) approach of graph transformations in the sense of [Ehr79, Roz97]. As net formalism we use P/T-systems following the notation of "Petri nets are Monoids" in [MM90]. In this notation a P/T-system is given by $PN = (P, T, pre, post, M^0)$ with pre- and post domain functions $pre, post : T \to P^{\oplus}$ and initial marking $M^0 \in P^{\oplus}$, where $P^{\oplus}$ is the free commutative monoid over the set P of places with binary operation $\oplus$. Note that M^0 can also be considered as function $M^0 : P \to \mathbb{N}$ with finite support and the monoid notation $M^0 = 2p_1 \oplus 3p_2$ means that we have two tokens on place p_1 and three tokens on p_2. A transition $t \in T$ is M-enabled for a marking $M \in P^{\oplus}$ if we have $pre(t) \leq M$ and in this case the follower marking M' is given by $M' = M \ominus pre(t) \oplus post(t)$. Note that the inverse $\ominus$ of $\oplus$ is only defined in $M_1 \ominus M_2$ if we have $M_2 \leq M_1$.

In order to define rules and transformations of P/T-systems we have to introduce P/T-morphisms which are suitable for our purpose.

Definition 1 (P/T-Morphisms).
Given P/T-systems $PN_i = (P_i, T_i, pre_i, post_i, M_i^0)$ *for* $i = 1, 2$, *a P/T-morphism* $f : PN_1 \to PN_2$ *is given by* $f = (f_P, f_T)$ *with functions* $f_P : P_1 \to P_2$ *and* $f_T : T_1 \to T_2$ *satisfying*

$$\textbf{(1)}\ f_P^{\oplus} \circ pre_1 = pre_2 \circ f_T \text{ and } f_P^{\oplus} \circ post_1 = post_2 \circ f_T$$
$$\textbf{(2)}\ f_P^{\oplus}(M_{1|p}^0) \leq M_{2|f_P(p)}^0 \text{ for } p \in P_1$$

Note that the extension $f_P^{\oplus} : P_1^{\oplus} \to P_2^{\oplus}$ *of* $f_P : P_1 \to P_2$ *is defined by* $f_P^{\oplus}(\sum_{i=1}^{n} k_i \cdot p_i) = \sum_{i=1}^{n} k_i \cdot f_P(p_i)$ *and the restriction* $M_{1|p}^0$ *by* $M_{1|p}^0 = M_1^0(p) \cdot p$ *where* M_1^0 *is considered as function* $M_1^0 : P \to \mathbb{N}$. *(1) means that* f *is compatible with pre- and post domain and (2) that the initial marking of* N_1 *at place* p *is smaller or equal to that of* N_2 *at* $f_P(p)$. *Moreover the P/T-morphism* f *is called strict if* $f_P^{\oplus}(M_{1|p}^0) = M_{2|f_P(p)}^0$ *and* f_P, f_T *are injective* **(3)**.

The category defined by P/T-systems and P/T-morphisms is denoted by **PTSys** where the composition of P/T-morphisms is defined componentwise for places and transitions. Examples of P/T-morphisms are given in Fig. 7.

The next step in order to define transformations of P/T-systems is to define the gluing of P/T-systems in analogy to concatenation in the string case.

Definition 2 (Gluing of P/T-Systems).
Given P/T-systems $PN_i = (P_i, T_i, pre_i, post_i, M_i^0)$ *for* $i = 0, 1, 2$ *with strict inclusion* $inc : PN_0 \to PN_1$ *and P/T-morphism* $f : PN_0 \to PN_2$. *Then the gluing* PN_3 *of* PN_1 *and* PN_2 *via* (PN_0, f), *written* $PN_3 = PN_1 +_{(PN_0, f)} PN_2$, *is defined by the following diagram (1), called "gluing diagram", with*

1. $\forall p \in P_1 = P_0 \uplus (P_1 \setminus P_0) : f'_P(p) = \underline{if}\ p \in P_0\ \underline{then}\ f_P(p)\ \underline{else}\ p$
   $\forall t \in T_1 = T_0 \uplus (T_1 \setminus T_0) : f'_T(t) = \underline{if}\ t \in T_0\ \underline{then}\ f_T(t)\ \underline{else}\ t$

2. $PN_3 = (P_3, T_3, pre_3, post_3, M_3^0)$ *with*
   - $P_3 = P_2 \uplus (P_1 \setminus P_0)$, $T_3 = T_2 \uplus (T_1 \setminus T_0)$,
   - $pre_3(t) =$ $\underline{if}$ $t \in T_2$ $\underline{then}$ $pre_2(t)$ $\underline{else}$ $f_P'^{\oplus}(pre_1(t))$,
   - $post_3(t) =$ $\underline{if}$ $t \in T_2$ $\underline{then}$ $post_2(t)$ $\underline{else}$ $f_P'^{\oplus}(post_1(t))$ *and*
   - $M_3^0 = M_2^0 \oplus (M_1^0 \ominus M_0^0)$.

$$\begin{array}{ccc} PN_0 & \xrightarrow{inc} & PN_1 \\ {\scriptstyle f}\downarrow & (1) & \downarrow{\scriptstyle f'} \\ PN_2 & \xrightarrow[inc']{} & PN_3 \end{array}$$

Remark 1. The disjoint union in the definition of P_3 and T_3 takes care of the problem that there may be places or transitions in PN_2, which are - by chance - identical to elements in $P_1 \setminus P_0$ or $T_1 \setminus T_0$, but only elements in PN_0 and $f(PN_0)$ should be identified. In this case the elements of $P_1 \setminus P_0$ and $T_1 \setminus T_0$ should be renamed before applying the construction above.

Proposition 1 (Gluing of P/T-Systems).
The gluing $PN_3 = PN_1 +_{(PN_0,f)} PN_2$ is a well-defined P/T-system such that $f' : PN_1 \to PN_3$ is a P/T-morphism, $inc' : PN_2 \to PN_3$ is a strict inclusion and the gluing diagram (1) commutes, i.e. $f' \circ inc = inc' \circ f$.

Proof. 1. *PN_3 is a well-defined P/T-system, because $pre_3, post_3 : T_3 \to P_3^{\oplus}$ are well-defined functions. Now $f' = (f_P', f_T') : PN_1 \to PN_3$ is a P/T-morphism, because we have $pre_3 \circ f_T' = f_P'^{\oplus} \circ pre_1$ (and similar for post) by case distinction:*

Case 1. For $t \in T_0$ we have $pre_3(f_T'(t)) = pre_3(f_T(t)) = pre_2(f_T(t)) =$
$$f_P^{\oplus}(pre_0(t)) = f_P'^{\oplus}(pre_0(t)) = f_P'^{\oplus}(pre_1(t)).$$

Case 2. For $t \in T_1 \setminus T_0$ we have $pre_3(f_T'(t)) = pre_3(t) = f_P'^{\oplus}(pre_1(t))$.

We have marking compatibility of f' by:

Case 1. For $p \in P_0$ we have
$$f_P'^{\oplus}(M^0_{1|p}) = f_P^{\oplus}(M^0_{0|p}) \leq M^0_{2|f_P(p)} \leq M^0_{3|f_P(p)} = M^0_{3|f_P'(p)}.$$

Case 2. For $p \in P_1 \setminus P_0$ we have
$$f_P'^{\oplus}(M^0_{1|p}) = f_P'^{\oplus}((M_1^0 \ominus M_0^0)_{|p}) = (M_1^0 \ominus M_0^0)_{|p} \leq M^0_{3|f_P'(p)} .$$

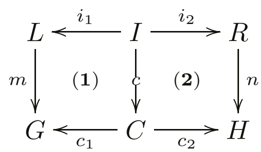
2. *$inc' : PN_2 \to PN_3$ is a P/T-system inclusion by construction. The marking M_3^0 is well-defined because $M_0^0 \leq M_1^0$ and $M^0_{0|p} = M^0_{1|p}$ for $p \in P_0$ by strict inclusion $inc : PN_0 \to PN_1$. Moreover inc' is strict, because we have $M_1^0 \ominus M_0^0 \in (P_1 \setminus P_0)^{\oplus}$ which implies for $p \in P_2$ $M^0_{2|p} = M^0_{3|p}$.*
3. *$f' \circ inc = inc' \circ f$ by construction*

Remark 2. The gluing diagram (1) is a pushout diagram in the category **PTSys**. This implies that the transformation of P/T-systems defined below is in the spirit of the double-pushout approach for graph transformations and high-level replacement systems (see [Ehr79, EHK91]).

Two examples of gluing and gluing diagrams are given in Fig. 7, where $G = L_2 +_{I_2} C$ and $H = R_2 +_{I_2} C$ in the left hand and the right hand gluing diagram respectively. Our next goal is to define rules, application of rules and transformations of P/T-systems.

Definition 3 (Rule of P/T-Systems). *A rule $r = (L \xleftarrow{i_1} I \xrightarrow{i_2} R)$ of P/T-systems consists of P/T-systems L, I, and R, called left-hand side, interface, and right-hand side of r respectively, and two strict P/T-morphisms $I \xrightarrow{i_1} L$ and $I \xrightarrow{i_2} R$ which are inclusions.*

Remark 3. The application of a rule r to a P/T-system G is given by a P/T-morphism $L \xrightarrow{m} G$, called match. Now a direct transformation $G \overset{r}{\Longrightarrow} H$ via r can be constructed in two steps. In a first step we construct the context C given by $(G - m(L)) \cup m \circ i_1(I)$ and P/T-morphisms $I \xrightarrow{c} C$ and $C \xrightarrow{c_1} G$, where c_1 is a strict inclusion. This means we remove the match $m(L)$ from G and preserve the interface $m \circ i_1(I)$. In order to make sure that C becomes a subsystem of G we have to require a "gluing condition" (see Def. 4). This makes sure that C is a P/T-system and we have $m \circ i_1 = c_1 \circ c$ in the "context diagram" (1). In the second step we construct H as gluing of C and R along I, this means we obtain the gluing diagram (2) from $I \xrightarrow{c} C$ and $I \xrightarrow{i_2} R$.



Now we define the gluing condition and the context construction.

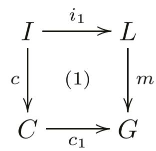
Definition 4 (Gluing Condition).
Given a strict inclusion morphism $i_1 : I \to L$ and a P/T-morphism $m : L \to G$ the gluing points GP, dangling points DP and the identification points IP of L are defined by

$$GP = P_I \cup T_I$$
$$DP = \{p \in P_L | \exists t \in (T_G \setminus m_T(T_L)) : m_P(p) \in pre_G(t) \oplus post_G(t)\}$$
$$IP = \{p \in P_L | \exists p' \in P_L : p \neq p' \wedge m_P(p) = m_P(p')\} \cup \{t \in T_L | \exists t' \in T_L : t \neq t' \wedge m_T(t) = m_T(t')\}$$

where $p \in P_L = \sum_{i=1}^{n} k_i \cdot p_i$ means $p = p_i$ and $k_i \neq 0$ for some i. Then the gluing condition is satisfied if all dangling and identifications points are gluing points, i.e $DP \cup IP \subseteq GP$.

Proposition 2 (Context P/T-System). *Given a strict inclusion $i_1 : I \to L$ and a P/T-morphism $m : L \to G$ then the following context P/T-system C is well-defined and leads to the following commutative diagram (1), called "context diagram", if the gluing condition $DP \cup IP \subseteq GP$ is satisfied.*

$C = (P_C, T_C, pre_C, post_C, M_C^0)$ is defined by
$P_C = (P_G \setminus m_P(P_L)) \cup m_P(P_I)$,
$T_C = (T_G \setminus m_T(T_L)) \cup m_T(T_I)$,
$pre_C = pre_{G|C}, post_C = pre_{G|C}$ *and*
$M_C^0 = M_{G|C}^0$.



The morphisms in (1) are defined by $c : I \to C$ to be the restriction of $m : L \to G$ to I, and $c_1 : C \to G$ to be a strict inclusion.

Proof. *The P/T-system C and $pre_C, post_C : T_C \to P_C^{\oplus}$ with $pre_C = pre_{G|C}$ and $post_C = pre_{G|C}$ are well-defined if $DP \cup IP \subseteq GP$. For $t \in T_C$ we have to show $pre_C(t) \in P_C^{\oplus}$ (and similar for $post_C(t)$).*

Case 1. For $t \in T_G \setminus m_T(T_L)$ we have $pre_C(t) = pre_G(t) = \sum_{i=1}^{n} k_i \cdot p_i$. Assume $p_i \notin P_C$ for some $i \leq n$. Then $p_i \in m_P(P_L) \setminus m_P(P_I)$ with $p_i \in pre_G(t)$. Hence there is $p'_i \in P_L \setminus P_I$ with $m_P(p'_i) = p_i$. This implies $p'_i \in DP$ and $p'_i \notin GP$ and contradicts the gluing condition $DP \cup IP \subseteq GP$.

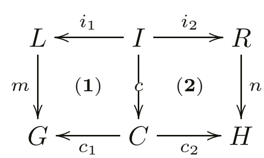
Case 2. For $t \in m_T(T_I)$ we have $t' \in T_I$ with $t = m_T(t')$. This implies $pre_C(t) = pre_G(t) = pre_G(m_T(t')) = m_P^{\oplus}(pre_L(t')) = m_P^{\oplus}(pre_I(t')) \in m_P^{\oplus}(P_I^{\oplus}) = (m_P(P_I))^{\oplus} \subseteq P_C^{\oplus}$.

Moreover $c : I \to C$ satisfies the marking condition (2) in Def. 1, because this is true for $m : L \to G$ and c is restriction of m. Finally $c_1 : C \to G$ is a strict inclusion by construction. This leads to the commutative diagram (1) in **PTSys**.

Remark 4. Note that we have not used the "identification condition" $ID \subseteq GP$, which is part of the gluing condition. But this is needed to show that the context diagram (1) is - up to isomorphism - also a gluing diagram and hence a pushout diagram in the category **PTSys**. This means that C is constructed in such a way that G becomes the gluing of L and C via I, i.e. $G \cong L +_I C$.

An example of a context diagram is the left diagram in Fig. 7, where C is the context P/T-system for $i_2 : I_2 \to L_2$ and $g : L_2 \to G$. Now a direct transformation is given by the combination of a context diagram and a gluing diagram.

Definition 5 (Applicability of Rules and Transformation). *A rule $r = (L \xleftarrow{i_1} I \xrightarrow{i_2} R)$ is called applicable at match $L' \xrightarrow{m} G$ if $L = L'$ and the gluing condition is satisfied for i_1 and m. In this case we obtain a context P/T-system C with context diagram (1) and a gluing diagram (2) with $H = C +_I R$ leading to a direct transformation $G \overset{r}{\Longrightarrow} H$ consisting of the following diagrams (1) and (2). A (rule-based) transformation $G \overset{*}{\Longrightarrow} H$ is a sequence of direct transformations $G = G_0 \overset{r_1}{\Longrightarrow} G_1 \overset{r_2}{\Longrightarrow} \ldots \overset{r_n}{\Longrightarrow} G_n = H$ with $G = H$ for $n = 0$. An example for a direct transformation is given in Fig. 7.*



Remark 5. As pointed out in Remark 2 and Remark 4 already the context diagram (1) and the gluing diagram (2) are pushout diagrams in the category **PTSys**. Hence a direct transformation $G \stackrel{r}{\Longrightarrow} H$ is given by the two pushouts (1) and (2), also called double pushout (DPO). In the DPO-approach of graph transformations (see [Ehr79]), high-level replacement systems [EHK91] and Petri net transformations [EP04] a direct transformation is defined by a DPO-diagram. For P/T-systems our definition is equivalent up to isomorphism to the existence of a DPO in the category **PTSys**.

4 High-Level Nets with Nets and Rules as Tokens

In this section we review the definition of algebraic high-level (AHL) nets in the notation of [EHP02] and [EM85] for algebraic specifications. Moreover we present a specific HLNR-System-SIG signature and algebra. Both are essential for our new notion of high-level net and rule (HLNR) systems in order to model high-level nets with nets and rules as tokens.

Definition 6 (Algebraic High-Level Net). *An algebraic high-level (AHL) net* $AN = (\mathrm{SP}, P_{AN}, T_{AN}, pre_{AN}, post_{AN}, cond_{AN}, type_{AN}, A)$ *consists of*

- *an algebraic specification* $\mathrm{SP} = (\Sigma, E; X)$ *with signature* $\Sigma = (S, OP)$*, equations* E*, and additional variables* X*;*
- *a set of places* P_{AN} *and a set of transitions* T_{AN}*;*
- *pre- and post conditions* $pre_{AN}, post_{AN} : T_{AN} \to (T_\Sigma(X) \otimes P_{AN})^\oplus$*;*
- *firing conditions* $cond_{AN} : T_{AN} \to \mathcal{P}_{fin}(Eqns(\Sigma; X))$*;*
- *a type of places* $type_{AN} : P_{AN} \to S$ *and*
- *a* (Σ, E)*-algebra* A

where the signature $\Sigma = (S, OP)$ *consists of sorts* S *and operation symbols* OP*,* $T_\Sigma(X)$ *is the set of terms with variables over* X*,* $(T_\Sigma(X) \otimes P_{AN}) = \{(term, p) | term \in T_\Sigma(X)_{type_{AN}(p)}, p \in P_{AN}\}$ *and* $Eqns(\Sigma; X)$ *are all equations over the signature* Σ *with variables* X*.*

Definition 7 (Firing Behavior of AHL-Nets). *A marking of an AHL-Net* AN *is given by* $M_{AN} \in CP^\oplus$ *where* $CP = (A \otimes P_{AN}) = \{(a, p) | a \in A_{type_{AN}(p)}, p \in P_{AN}\}$*.*

The set of variables $Var(t) \subseteq X$ *of a transition* $t \in T_{AN}$ *are the variables of the net inscriptions in* $pre_{AN}(t), post_{AN}(t)$ *and* $cond_{AN}(t)$*. Let* $v : Var(t) \to A$ *be a variable assignment with term evaluation* $v^\sharp : T_\Sigma(Var(t)) \to A$*, then* (t, v) *is a consistent transition assignment iff* $cond_{AN}(t)$ *is validated in* A *under* v*. The set* CT *of consistent transition assignments is defined by* $CT = \{(t, v) | (t, v) \text{ consistent transition assignment}\}$*.*

A transition $t \in T_{AN}$ *is enabled in* M_{AN} *under* v *iff* $(t, v) \in CT$ *and* $pre_A(t, v) \leq M_{AN}$*, where* $pre_A : CT \to CP^\oplus$ *defined by* $pre_A(t, v) = \hat{v}(pre(t)) \in (A \otimes P_{AN})^\oplus$ *and* $\hat{v} : (T_\Sigma(Var(t)) \otimes P_{AN})^\oplus \to (A \otimes P_{AN})^\oplus$ *is the* *obvious extension of* $v^\sharp$ *to terms and places (similar* $post_A : CT \to CP^\oplus$*). Then the follower marking is computed by* $M'_{AN} = M_{AN} \ominus pre_A(t,v) \oplus post_A(t,v)$.

The marking graph MG *of* AN *consists of all markings* $M \in CP^\oplus$ *as nodes and all* $M_{AN} \xrightarrow{(t,v)} M'_{AN}$ *as edges where* M'_{AN} *is the follower marking of* M_{AN} *provided that* t *is enabled in* M_{AN} *under* v *with* $(t,v) \in CT$*. For an initial marking* $INIT$ *of* AN *the reachability graph* RG *is the subgraph of* MG *reachable from* $INIT$*.*

In order to allow P/T-systems and rules as tokens of an AHL-net AN we provide a specific specification SP and SP-algebra A based on the construction in the previous section. In fact, it is sufficient to consider as specific SP a signature, called HLNR-System-SIG, together with a suitable HLNR-System-SIG-algebra A, where HLNR-System refers to high-level net and rule systems.

Definition 8 (HLNR-System-SIG Signature and Algebra).
Given vocabularies T_0 *and* P_0*, the signature* HLNR-System-SIG *is given by*

HLNR-System-SIG =
sorts: $Transitions, Places, Bool, System, Mor, Rules$
opns: $tt, ff: \to Bool$
$enabled : System \times Transitions \to Bool$
$fire : System \times Transitions \to System$
$applicable : Rules \times Mor \to Bool$
$transform : Rules \times Mor \to System$
$coproduct : System \times System \to System$
$isomorphic : System \times System \to Bool$
$cod : Mor \to System$

and the HLNR-System-SIG*-algebra* A *for P/T-systems and rules as tokens is given by*

- $A_{Transitions} = T_0, A_{Places} = P_0, A_{Bool} = \{true, false\}$,
- A_{System} *the set of all P/T-systems over* T_0 *and* P_0*, i.e.*
  $A_{System} = \{PN | PN = (P,T,pre,post,M) \text{ P/T-system}, P \subseteq P_0, T \subseteq T_0\} \cup \{undef\}$,
- A_{Mor} *the set of all P/T-morphisms for* A_{System}*, i.e.*
  $A_{Mor} = \{f | f : PN \to PN' \text{ P/T-morphism with } PN, PN' \in A_{System}\}$,
- A_{Rules} *the set of all rules of P/T-systems, i.e.*
  $A_{Rules} = \{r | r = (L \xleftarrow{i_1} I \xrightarrow{i_2} R) \text{ rule of P/T-systems with strict inclusions } i_1, i_2\}$,
- $tt_A = true$, $ff_A = false$,
- $enabled_A : A_{System} \times T_0 \to \{true, false\}$ *for* $PN = (P,T,pre,post,M)$ *with*

$$enabled_A(PN,t) = \begin{cases} true & \text{if } t \in T, pre(t) \leq M \\ false & \text{else} \end{cases}$$

- $fire_A : A_{System} \times T_0 \to A_{System}$ *for* $PN = (P,T,pre,post,M)$ *with*

$$fire_A(PN,t) = \begin{cases} (P,T,pre,post,M \ominus pre(t) \oplus post(t)) & \\ & \text{if } enabled_A(PN,t) = tt \\ undef & else \end{cases}$$

– $applicable_A : A_{Rules} \times A_{Mor} \to \{true, false\}$ *with*

$$applicable_A(r,m) = \begin{cases} true & \text{if } r \text{ is applicable at match } m \\ false & else \end{cases}$$

– $transform_A : A_{Rules} \times A_{Mor} \to A_{System}$ *with*

$$transform_A(r,m) = \begin{cases} H & \text{if } applicable_A(r,m) \\ undef & else \end{cases}$$

where for $L \xrightarrow{m} G$ *and* $applicable_A(r,m) = true$ *we have a direct transformation* $G \xRightarrow{r} H$,

– $coproduct_A : A_{System} \times A_{System} \to A_{System}$ *the disjoint union (i.e. the two P/T-systems are combined without interaction) with*

$$coproduct_A(PN_1,PN_2) = \underline{if}\ (PN_1 = undef \vee PN_2 = undef)\ \underline{then}\ undef$$
$$\underline{else}\ ((P_1 \uplus P_2),(T_1 \uplus T_2), pre_3, post_3, M_1 \oplus M_2)$$

where $pre_3, post_3 : (T_1 \uplus T_2) \to (P_1 \uplus P_2)^{\oplus}$ *are defined by*
$pre_3(t) = \underline{if}\ t \in T_1\ \underline{then}\ pre_1(t)\ \underline{else}\ pre_2(t)$
$post_3(t) = \underline{if}\ t \in T_1\ \underline{then}\ post_1(t)\ \underline{else}\ post_2(t)$

– $isomorphic_A : A_{System} \times A_{System} \to \{true, false\}$ *with*

$$isomorphic_A(PN_1,PN_2) = \begin{cases} true & \text{if } PN_1 \cong PN_2 \\ false & else \end{cases}$$

where $PN_1 \cong PN_2$ *means that there is a strict P/T-morphism* $f = (f_P, f_T) : PN_1 \to PN_2$ *s.t.* f_P, f_T *are bijective functions,*

– $cod_A : A_{Mor} \to A_{System}$ *with* $cod_A\ (f : PN_1 \to PN_2) = PN_2$.

Definition 9 (High-Level Net and Rule Systems).
Given the signature HLNR-System-SIG *and the* HLNR-System-SIG*-algebra* A *as above, a high-level net and rule system* $HLNR = (AN, INIT)$ *consists of an AHL-net* AN *(see Def. 6) with* SP = (HLNR-System-SIG; X) *where* X *are variables over* HLNR-System-SIG, *and initial marking* $INIT$ *of* AN *such that*

1. *all places* $p \in P_{AN}$ *are either*
   - system places i.e. $p \in P_{Sys} = \{p \in P_{AN} | type_{AN}(p) = System\}$ *or*
   - rule places i.e. $p \in P_{Rules} = \{p \in P_{AN} | type_{AN}(p) = Rules\}$,
2. *all rule places* $p \in P_{Rules}$ *are contextual, i.e. for all transitions* $t \in T_{AN}$ *connected with* p *there exists a variable* $r \in X$ *such that* $pre_{AN}(t)_{|p} = post_{AN}(t)_{|p} = r$, *i.e. in the net structure of* AN *the connection between* p *and* t *is given by a double arrow labeled with the variable* r.

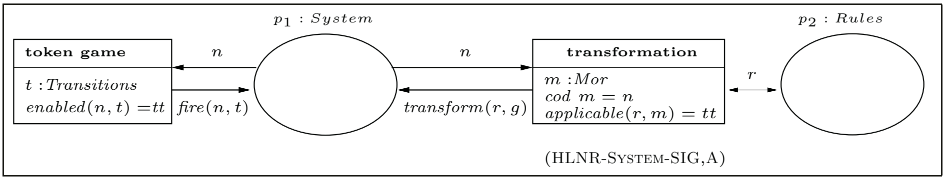


Fig. 8. Basic high-level net and rule system

Remark 6. Our notion of HLNR-systems has static rules. This means that the tokens representing our rules do not move and remain unchanged on the rule places (see Section 6 for extensions). According to our paradigm "nets and rules as tokens" we only allow system and rule places, but no places which are typed by other sorts of HLNR-System-SIG. A HLNR-system with only one system place and one rule place is called basic HLNR-system.

Example 1 (Basic HLNR-System). A basic HLNR-system with system place p_1 and rule place p_2 is shown in Fig. 8 where the empty initial marking can be replaced by suitable P/T-systems resp. rules on these places.

Example 2 (House of Philosophers). In Section 2 we have given a detailed discussion of the HLNR-system "House of Philosophers" as given in Fig. 1 with system places *Library, Entrance-Hall*, and *Restaurant* and rule places $Rule_1, \ldots, Rule_4$.

The behavior of a HLNR-system $HLNR = (AN, INIT)$ is given by the reachability graph in the sense of AHL-nets (see Def. 7), but it can be represented more explicitly as follows:

Proposition 3 (Reachability Graph of High-Level Net and Rule System). *The reachability graph RG of a HLNR-system $HLNR = (AN, INIT)$ can be characterized as follows:*

1. *Each node of RG is represented by a system family $F \in (A_{System} \times P_{Sys})^{\oplus}$ i.e. $F = \sum_{i=1}^{n}(PN_i, p_i)$ with $PN_i \in A_{System}$ and $p_i \in P_{Sys}$;*
2. *Each edge of RG is represented by $F \overset{(t_{AN},v)}{\longrightarrow} F'$, where $(t_{AN}, v) \in CT_{AN}$ is a consistent transition assignment.*

A system family $F = \sum_{i=1}^{n}(PN_i, p_i)$ is well-formed if $PN_i \neq undef$ for all $i = 1, \ldots, n$. If the system family of $INIT$ is well-formed and all $(t_{AN}, v) \in CT_{AN}$ of RG are strongly consistent, i.e. all terms of sort System in $pre_{AN}(t_{AN}), post_{AN}(t_{AN})$ and $cond_{AN}(t_{AN})$ are evaluated under $v^{\sharp}$ to defined elements $PN \neq undef$, then we have:

3. *The reachability graph RG is well-formed, i.e. the system families of all nodes of RG are well-formed.*

Proof. Each node of RG is given by a marking $M_{AN} \in (A \otimes P_{AN})^{\oplus}$, i.e. $M_{AN} = \sum_{i=1}^{n}(a_i, p_i)$ with $p_i \in P_{AN}$ and $a_i \in A_{type(p)}$. Since we have $P_{AN} = P_{Sys} \cup$

P_{Rules} and all rule places are contextual the restriction M_{Rules} of M_{AN} to all $p_i \in P_{Rules}$ is the same for all markings and represents the token rules on the rule places in the initial marking $INIT$. This means that each M_{AN} is uniquely represented by the restriction M_{Sys} of M_{AN} to all $p_i \in P_{Sys}$, w.l.o.g. $M_{Sys} = \sum_{i=1}^{n'}(a_i, p_i)$ with $n' \leq n$ and $p_i \in P_{Sys}, a_i \in A_{System}(i = 1, \ldots, n')$. This means $M_{Sys} \in (A_{System} \times P_{Sys})^{\oplus}$. Hence each M_{AN} of RG is represented by the system family M_{Sys} and each edge $M_{AN} \overset{(t_{AN},v)}{\longrightarrow} M'_{AN}$ by $M_{Sys} \overset{(t_{AN},v)}{\longrightarrow} M'_{Sys}$. If $INIT_{Sys}$ is well-formed then for each $M_{Sys} \overset{(t_{AN},v)}{\longrightarrow} M'_{Sys}$ with well-formed M_{Sys} strong consistency of (t_{AN}, v) implies that also M'_{Sys} is well-formed. This implies that the reachability graph RG is well-formed.

Remark 7. Strong consistency of $(t_{AN}, v) \in CT_{AN}$ can be achieved for a HLNR-system $HLNR$ by including equations of the form $enabled(n,t) = tt$ or $applicable(r,m) = tt$ into $cond_{AN}(t_{AN})$ as shown in Fig. 1 and Fig. 8.

An interesting special case of HLNR-systems are basic HLNR-systems as presented in Fig. 8 of Example 1. Let us assume that the initial marking is given by a P/T-system PN on place p_1 and a set $RULES$ of token rules on place p_2. Then $(PN, RULES)$ can be considered as *reconfigurable P/T-system* in the following sense: on the one hand we can apply the token game and on the other hand rule-based transformations of the net structure of PN. Moreover these activities can be interleaved. This allows to model changes of the net structure while the system is running. This is most important for changes on the fly of large systems, where it is important to keep the system running, while changes of the structure of the system have to be applied. It would be especially important to analyze under which conditions the token game activities are independent of the transformations. This problem is closely related to local Church-Rosser properties for graph resp. net transformations, which are valid in the case of parallel independence of transformations (see [Ehr79, EP04]).

5 Specification and Implementation Aspects

In the previous section we have presented an explicit version of HLNR-systems based on AHL-nets. The main idea was to present a set theoretical version of the HLNR-SYSTEM-SIG-algebra A which defines our concept of "nets and rules as tokens". For various reasons it is also interesting to present an algebraic specification of this algebra. Unfortunately first-order algebraic specifications in the sense of [EM85] or CASL [CAS94] are not suitable for this purpose. Actually we need higher-order features which are provided by HASCASL [SM02], a higher-order extension of the common algebraic specification language CASL.

HASCASL-specifications combine the simplicity of algebraic specifications with higher-order features including function types. It is geared towards specification of functional programs, in particular in Haskell. The semantics of HASCASL is defined by a set-theoretic notion of intensional algebras. The advantage is that in an intensional setting function equivalence testing is possible within some models. Moreover, we can distinguish between different functions that exhibit the same behavior. By contrast extensional equality of functions means that two functions are equal if they always produce the same results for the same arguments. Standard ML, the data type part of Coloured Petri (CP) nets [Jen92], cannot implement equality on function types. This means that it would be difficult to consider P/T-systems and rules as defined in Section 3 as first-class citizens and thus tokens in CP-nets. In our technical report [HM04] we have presented a HasCasl-specification of P/T-systems, P/T-morphism and of rule-based transformations according to the definitions in Section 3. This leads to the formalism of algebraic higher-order (AHO) nets [HM03] where in contrast to AHL-nets higher-order algebraic specifications in HasCasl are used. Since tools for HasCasl already have been implemented this is a first step towards an implementation of our approach presented in this paper.

In fact several aspects of HLNR-systems are supported by tools. The algebraic approach to graph transformations which can also be used for rule-based transformations of nets, is supported by the graph transformation environment AGG (see the homepage of [AGG]). AGG includes an editor for graphs and graph grammars, a graph transformation engine, and a tool for the analysis of graph transformations. On top of the graph transformation system AGG there is the GenGED environment (see the homepage of [Gen]) that supports the generic description of visual modeling languages for the generation of graphical editors and the simulation of the behavior of visual models. Especially, rule-based transformations for P/T-systems can be expressed using GenGED. These transformations can be coupled to other Petri net tools using the Petri Net Kernel [KW01], a tool infrastructure for editing, simulating, and analyzing Petri nets of different net classes and for integration of other Petri net tools. On the level of the data type part the Heterogeneous Tool Set (Hets) (see the homepage of [Hets]) provides a parser and static analysis for Casl and HasCasl-specifications; theorem proving support in form of a translation to the Isabelle/HOL prover is under development. Also, a translation tool from a HasCasl subset to Haskell is provided.

6 Conclusion and Future Work

In this paper we have presented the new concept of high-level nets with rules and nets as tokens and initial marking, short HLNR-systems, which realizes our new paradigm of "nets and rules as tokens". This extends Valk's paradigm "nets as tokens" and also partly his notion of elementary object systems [Val98, Val01]. In Section 2 we have presented a detailed case study of the "House of Philosophers", which allows to give an example driven introduction to HLNR-systems. Moreover we have discussed the relationship to other approaches and pointed out that it seems to be useful and possible to extend our approach by object-oriented features and also to an interaction relation in the sense of Valk.

In Section 3 we have presented the main concepts for our paradigm "nets and rules as tokens". Due to the net inscriptions a firing step in the system level realizes on the one hand the computation of the follower marking of a net (i.e. a P/T-system) and on the other hand the modification of a net by an appropriate rule. Thus transformations become effectively included in the system enabling the system to transform nets as tokens in a formal way by using also rules as tokens. For this purpose we have introduced rule-based transformations for P/T-systems in this paper. In fact we have presented an explicit version of transformations avoiding pushout constructions, but our approach is equal - up to isomorphism - to a double-pushout (DPO) approach in the sense of [Ehr79, EHK91], which will allow to obtain also several other results already known for the DPO-approach [Roz97]. From this point of view the paper presents an interesting integration of concepts in the area of graph transformations and Petri nets.

In HLNR-systems the coupling of a set of rules as tokens to certain transitions is fixed due to the given net topology. In future work we will consider also the migration of rules as tokens. This means the mechanism of mobility and modification presented in our example could be transferred to rules as tokens in order to achieve even more expressive models. The mobility aspect of rules as tokens can be easily introduced by further transitions connecting places of the type *Rules*. However the modification of rules as tokens (see [PP01]) requires an extension of the corresponding algebra in Section 4.

Another interesting aspect for future work is to study transformations of P/T-systems which preserve properties like safety or liveness. Especially in the area of workflow modeling the notion of soundness (which comprises liveness) is of importance (see e.g. [Aal98]). Here we can use the approach of property preserving rules (see [PU03] for an overview). To integrate these kinds of rules into HLNR-systems the set of rules A_{Rules} of the HLNR-System-Sig-algebra A (see Section 4) would have to be restricted to property preserving rules.

Finally in Section 5 we have presented several specification and implementation aspects which are useful towards tool-support for our new concepts.

References

[Aal98] W.M.P. van der Aalst. The Application and Theory of Petri Nets to Workflow Management Systems. *The Journal of Circuits, Systems and Computers*, 8:21-66. 1998.

[Aal02] W.M.P. van der Aalst. Inheritance of Workflows: An Approach to Tackling Problems Related to Change. *Theoretical Computer Science*, 270(1-2):125–203. 2002.

[AGG] AGG Homepage. http://tfs.cs.tu-berlin.de/agg.

[ADC01] G. Agha, F. De Cindio, and G. Rozenberg, editors. *Concurrent Object-Oriented Programming and Petri Nets*, *LNCS* 2001. Springer, 2001.

[CAS94] Mosses, Peter D. *CASL Reference Manual - The Complete Documentation of the Common Algebraic Specification Language*, LNCS 2960. Springer, 2004.

[Ehr79] H. Ehrig. Introduction to the algebraic theory of graph grammars (A survey). In *Graph Grammars and their Application to Computer Science and Biology*, LNCS 73, pages 1–69. Springer, 1979.

[EHK91] H. Ehrig, A. Habel, H.-J. Kreowski, and F. Parisi-Presicce. Parallelism and concurrency in high-level replacement systems. *Math. Struct. in Comp. Science*, 1:361–404, 1991.

[EM85] H. Ehrig, B. Mahr. *Fundamentals of Algebraic Specifications 1: Equations and Initial Semantics*. Springer Verlag, EATCS Monographs in Theoretical Computer Science, 1992.

[EHP02] H. Ehrig, K. Hoffmann, J. Padberg, P. Baldan, and R. Heckel. High Level Net Processes. In *Formal and Natural Computing*, LNCS 2300, pages 191-219. Springer, 2002.

[EP04] H. Ehrig and J. Padberg. Graph Grammars and Petri Net Transformations. In *Lectures on Concurrency and Petri Nets. Special Issue Advanced Course PNT*, LNCS 3098, pages 496-536. Springer, 2004.

[Far99] B. Farwer A Linear Logic View of Object Petri Nets. *Fundamenta Informaticae, IOS Press*, 37:225-246, 1999.

[Gen] GenGED Homepage. http://tfs.cs.tu-berlin.de/genged.

[Hets] Hets Homepage. http://www.tzi.de/cofi/hets.

[Hof00] K. Hoffmann. Run Time Modification of Algebraic High Level Nets and Algebraic Higher Order Nets using Folding and Unfolding Constructions. In G. Hommel, editor, *Proc. of 3th Int. Workshop of Communication Based Systems*, pages 55-72. Kluwer, 2000.

[HM03] K. Hoffmann and T. Mossakowski. Algebraic Higher-Order Nets: Graphs and Petri Nets as Tokens. In M. Wirsing, D. Pattinson, and R. Henicker, editors, *Proc. of 16th Int. Workshop of Algebraic Development Techniques*, LNCS 2755, pages 253–267. Springer, 2003.

[HM04] K. Hoffmann and T. Mossakowski. Integration of Petri nets and Rule-Based Transformations using Algebraic Higher-Order Nets. Technical Report, Technical University of Berlin, 2004.

[Jen92] K. Jensen. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use*, volume 1: Basic Concepts. Springer Verlag, EATCS Monographs in Theoretical Computer Science, 1992.

[KW01] E. Kindler and M. Weber. The Petri net kernel – an infrastructure for building Petri net tools. *Software Tools for Technology Transfer*, 3(4):486–497, 2001.

[MM90] J. Meseguer and U. Montanari. Petri Nets are Monoids. *Information and Computation*, 88(2):105–155, 1990.

[MTH97] R. Milner, M. Tofte, R. Harper, and D. MacQueen. *The Definition of Standard ML - Revised*. MIT Press, 1997.

[Mos05] T. Mossakowski. Heterogeneous specification and the heterogeneous tool set. Habilitation thesis, University of Bremen, 2004.

[PER95] J. Padberg, H. Ehrig, and L. Ribeiro. Algebraic high-level net transformation systems. *Mathematical Structures in Computer Science*, 5:217–256, 1995.

[PU03] J. Padberg, and M. Urbasek. Rule-Based Refinement of Petri Nets: A Survey. In *Advances in Petri Nets: Petri Net Technologies for Modeling Communication Based Systems*, LNCS 2472, pages 161–196. Springer, 2003.

[PP01] Francesco Parisi-Presicce. On modifying high level replacement systems. In Hartmut Ehrig, Claudia Ermel, and Julia Padberg, editors, *Electronic Notes in Theoretical Computer Science*, volume 44. Elsevier, 2001.

[RWL03] A. Ratzer, L. Wells, H. Lassen, M. Laursen, J. Qvortrup, M. Stissing, M. Westergaard, S. Christensen, K. Jensen. CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In *Proc. of the 24th Int. Conference on Applications and Theory of Petri Nets (ICATPN 2003)*, pages 450–462, LNCS 2679. Springer, 2003.

[Roz97] G. Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. World Scientific, 1997.

[SM02] L. Schröder and T. Mossakowski. HasCasl: Towards integrated specification and development of Haskell programs. In H. Kirchner and C. Reingeissen, editors, *Algebraic Methodology and Software Technology, 2002*, LNCS 2422, pages 99–116. Springer-Verlag, 2002.

[Sil01] C. Silbertin-Blanc. The Hurried Philosophers. In G. Agha, F. De Cindio, and G. Rozenberg, editors, *Concurrent Object-Oriented Programming and Petri Nets*, LNCS 2001, pages 536–537. Springer, 2001.

[Val98] Rüdiger Valk. Petri Nets as Token Objects: An Introduction to Elementary Object Nets. *Proc. of the International Conference on Application and Theory of Petri Nets*, LNCS 1420, pages 1–25, 1998.

[Val01] R. Valk. Concurrency in Communicating Object Petri Nets. In G. Agha, F. de Cindio, and G. Rozenberg, editors, *Concurrent Object-Oriented Programming and Petri Nets*, LNCS 2001, pages 164–195. Springer, 2001.

[Wie01] F. Wienberg. *Informations- und prozeorientierte Modellierung verteilter Systeme auf der Basis von Feature-Structure-Netzen*. PhD thesis, University Hamburg, 2001.