Plans as Formulae with a Non-commutative Logical Operator

Planning as Concurrency

Ozan Kahramanoğulları

University of Leipzig, Augustusplatz 10-11, 04109 Leipzig, Germany

Abstract. System NEL is a conservative extension of multiplicative exponential linear logic with a self-dual, non-commutative operator. In this paper, we express plans as logical formulae by using this sequential operator. We present an encoding of the conjunctive planning problems in the language of NEL where plans are not extracted from the proofs, but become explicit premises of derivations. We then extend the notion of a plan to cover parallel composition by employing a commutative logical operator. This way, like in concurrency, sequential and parallel composition come to the same level where the reasoning is done in a purely logical way. We provide a constructive correctness proof and an implementation of system NEL in Maude, and argue that this work is the primary step for providing a common language for planning and concurrency.

1 Introduction

Concurrency and planning are two fields of computer science that evolved independently, aiming at solving tasks that are similar in nature, but different in perspective: while planning formalisms focus on finding a plan (process) that solves a planning problem, the focus in process algebras, such as CCS [17], is concepts like dead-lock freeness and verification of security protocols. In such a perspective, a notion of equivalence of processes, e.g., bisimilarity, which respects the parallel behavior of the processes gains importance.

In a process algebra parallel and sequential composition are at the same level, since they are equivalently important notions for expressing concurrent processes. On the other hand, in planning the emphasis in the literature is on the sequential composition. In the case of the partial order planners, the plan is computed by heuristic methods, in a way distant from the underlying logical framework, and then the computed plan is linearized to be executed. However, parallel composition is natural in logic due to the commutative logical operators. In this paper, in a resource conscious setting, we establish the first, but crucial, step for our long term goal of a common language for concurrency and planning, inside a uniform deductive formalism.

With respect to resource consciousness, the relation between logic, actions and causality has been studied by various authors: In [1], Bibel imposes a syntactical condition called linearity on proofs, which requires that each literal is engaged in at most one connection. In [12], based on multiset rewriting, Hölldobler

and Schneeberger introduce an equational Horn logic where states are represented by an AC1 function symbol. In [16], Masseron *et al.* applies multiplicative fragment of Girard's linear logic [6] to resource conscious planning by axiomatizing the actions as proper axioms. Linear logic approach to planning is studied further by various authors [13, 15, 5]. In [7], it is shown to be equivalent to the approaches in [1] and [12].

In this paper, we further elaborate on the linear logic approach to planning, aiming at providing a common language for planning and concurrency. For this purpose we employ system NEL of *the calculus of structures* [9]. The calculus of structures is a proof theoretical formalism, which is a generalization of the one-sided sequent calculus with the gain of interesting proof theoretical properties. It was conceived to represent the logical system BV, which is a conservative extension of multiplicative linear logic with a self-dual, non-commutative operator, called *seq*. Tiu showed in [21] that this system can not be designed in the sequent calculus.

System NEL [10] is an extension of system BV with the exponentials of linear logic. In [2], Bruscoli showed that there is a correspondence between system BV and a fragment of CCS: the sequential composition corresponds to the non-commutative operator *seq*. Parallel composition is naturally mapped to the commutative linear logic operator *par*. However, as it is the case in CCS, there only the actions (labels) are included in the language, but not the resources that are consumed and produced by the actions.

In the following, we present an encoding of the conjunctive (multiset rewriting) planning problems in the language of NEL, where plans are not extracted from the proof of a planning problem, but become explicit premises of derivations. This way, similar to [2], by exploiting the non-commutative operator of system NEL, and the commutative logical operator *par*, we are able to observe concurrent plans, where the parallelism between plans is respected. Since our encoding is propositional, no unification mechanism is needed. This allows system NEL to give the complete operational semantics of our method, and establish the first step of a uniform formalism that connects concurrency and planning. This way, it becomes possible to transfer methods from concurrency to planning.

In [14], we presented an implementation of the system BV in Maude 2 [3]. By extending those modules by accommodating exponentials, in this paper, we also present an implementation of system NEL, where search for proofs and derivations becomes possible.

The rest of the paper is organized as follows: we begin with recapitulating notions and notations of conjunctive planning problems and system NEL. We then present an encoding of the conjunctive planning problems in the language of NEL and show that our encoding is correct for plans that are sequences of actions. Following this, we extend our correctness result to plans which include parallel composition. After presenting a sketch of a Maude 2 implementation of system NEL, we conclude with a brief discussion.

2 Planning Problems

Following [7, 16], a *planning domain* is given by: (1) a set of constants representing atomic properties of the world which we call *fluents* and denote by small letters; (2) a set of *transition rules (actions)* [1] that are multiset[2] rewrite rules; (3) *states* which are multisets of fluents. A conjunctive planning problem $\mathscr{P}$ is then given by $\langle \mathcal{I}, \mathcal{G}, \mathscr{A}, \mathcal{F} \rangle$ where $\mathcal{I} : \{\!| r_1, \ldots, r_m |\!\}$ is a multiset of fluents called *initial state*. The multiset $\mathcal{G} : \{\!| g_1, \ldots, g_n |\!\}$ of fluents is the *goal state*. $\mathscr{A}$ is a finite set of *actions* of the form $a : \{\!| c_1, \ldots, c_p |\!\} \rightarrow \{\!| e_1, \ldots, e_q |\!\}$, where $\{\!| c_1, \ldots, c_p |\!\}$ and $\{\!| e_1, \ldots, e_q |\!\}$ are multisets of fluents called *conditions* and *effects*, respectively, and a is the *name* of the action. $\mathcal{F} = \{f_1, \ldots, f_h\}$ is the set of all the fluents that appear in $\mathcal{I}$, $\mathcal{G}$ and $\mathscr{A}$.

An action is applicable in a state $\mathcal{S}$ iff $\{\!| c_1, \ldots, c_p |\!\} \dot{\subseteq} \mathcal{S}$. The application of an action a to a state $\mathcal{S}$ is defined by the function Φ as follows.

$$\Phi(a, \mathcal{S}) = (\, \mathcal{S} \dot{-} \{\!| c_1, \ldots, c_p |\!\}\,) \,\dot{\cup}\, \{\!| e_1, \ldots, e_q |\!\} \quad .$$

A goal $\mathcal{G}$ is satisfied iff there is a *plan (structure)* p, i.e., a sequence of actions $p = \langle a_1; \ldots; a_k \rangle$, which transforms the initial state into a state $\mathcal{S}$, i.e., $\Phi(a_k, \ldots, \Phi(a_1, \mathcal{I}) \ldots) = \mathcal{S}$ such that $\mathcal{G} \dot{\subseteq} \mathcal{S}$. If there exists such a plan p, then p is a solution for the planning problem $\mathscr{P}$. Then we say p *solves* $\mathscr{P}$. We denote the empty plan with $\circ$. If it is more convenient, $\Phi(a_k, \ldots, \Phi(a_1, \mathcal{I}) \ldots)$ will be abbreviated with $\Phi(p, \mathcal{I})$. The *length of a plan* is the number of actions in that plan.

Now, to illustrate the above theory on a planning problem, let us look at the following example which is a modification of an example from [7]. Suppose Bert is thirsty and wants to get some lemonade (l) from a vending machine. The lemonade costs 50 cents (f). Bert has a dollar bill (d) in his pocket. Because the vending machine accepts only 50 cents coins, Bert has to get change for his dollar. The problem of getting the lemonade can be described as a planning problem with the initial state $\mathcal{I} : \{\!| d |\!\}$, the actions $c_d : \{\!| d |\!\} \rightarrow \{\!| f, f |\!\}$ and $b_l : \{\!| f |\!\} \rightarrow \{\!| l |\!\}$ that allow him to change a dollar for two 50 cents coins, and to buy a lemonade, respectively. The goal state in which Bert got the lemonade is given by $\mathcal{G} : \{\!| l |\!\}$.

Clearly, the solution to the problem is the plan in which at first Bert changes the dollar and then buys the lemonade: applying this plan to the initial state yields, first, the state $\{\!| f, f |\!\}$, and then $\{\!| f, l |\!\}$. As the goal is contained in the last state, the planning problem is solved.

---

[1] We consider only propositional actions.

[2] Multisets are denoted by the curly brackets "{|" and "|}". $\dot{\cup}$, $\dot{-}$ and $\dot{\subseteq}$ denote the multiset operations corresponding to the usual set operations $\cup$, $-$ and $\subseteq$, respectively.

Associativity

$$[[R,T],U] = [R,[T,U]]$$
$$((R,T),U) = (R,(T,U))$$
$$\langle\langle R;T\rangle;U\rangle = \langle R;\langle T;U\rangle\rangle$$

Commutativity

$$[R,T] = [T,R]$$
$$(R,T) = (T,R)$$

Singleton

$$[R] = (R) = \langle R\rangle = R$$

Units

$$[\circ,R] = R$$
$$(\circ,R) = R$$
$$\langle\circ;R\rangle = R$$
$$\langle R;\circ\rangle = R$$

Exponentials

$$??R = ?R$$
$$!!R = !R$$
$$?\circ = \circ$$
$$!\circ = \circ$$

Negation

$$\bar{\circ} = \circ$$
$$\overline{[R,T]} = (\overline{R},\overline{T})$$
$$\overline{(R,T)} = [\overline{R},\overline{T}]$$
$$\overline{\langle R;T\rangle} = \langle\overline{R};\overline{T}\rangle$$
$$\overline{?R} = !\overline{R}$$
$$\overline{!R} = ?\overline{R}$$
$$\overline{\overline{R}} = R$$

Contextual Closure

if $R = T$ then $S\{R\} = S\{T\}$

Fig. 1. The equational system underlying System NEL.

3 The Calculus of Structures and System NEL

In this section, we present the calculus of structures [9] and system NEL [10] which is a conservative extension of multiplicative exponential linear logic with a non-commutative operator.

There are countably many *atoms*, denoted by $a, b, c, \ldots$ The *structures*[3] of the language NEL are denoted by P,Q,R, $S\ldots$ and are generated by

$$R ::= a \mid \circ \mid [\underbrace{R,\ldots,R}_{>0}] \mid (\underbrace{R,\ldots,R}_{>0}) \mid \langle\underbrace{R;\ldots;R}_{>0}\rangle \mid !R \mid ?R \mid \bar{R} \quad ,$$

where a stands for any atom and $\circ$, the *unit*, is not an atom. A structure $[R_1,\ldots,R_h]$ is a *par structure*, $(R_1,\ldots,R_h)$ is a *times structure*, $\langle R_1;\ldots;R_h\rangle$ is a *seq structure*, $!\,R$ is called an *of-course structure*, and $?\,R$ is called a *why-not structure*; $\bar{R}$ is the *negation* of the structure R. Structures are considered to be equivalent modulo the relation $=$, which is the smallest congruence relation induced by the equations shown in Figure 1. A *structure context*, denoted as in $S\{\ \}$, is a structures with a hole that does not appear in the scope of negation. The structure R is a *substructure* of $S\{R\}$ and $S\{\ \ \}$ is its *context*. Context braces are omitted if no ambiguity is possible.

In the calculus of structures, an *inference rule* is a scheme of the kind $\rho\,\dfrac{T}{R}$, where ρ is the *name* of the rule, T is its *premise* and R is its *conclusion*. A typical (deep) inference rule has the shape $\rho\,\dfrac{S\{T\}}{S\{R\}}$ and specifies a step of rewriting, by

[3] The notion of a structure is similar to the notion of a formula or a sequent of the sequent calculus. However, a structure denotes an equivalence class of structures. Fore a formal elaboration of this notion, we refer the reader to [9].

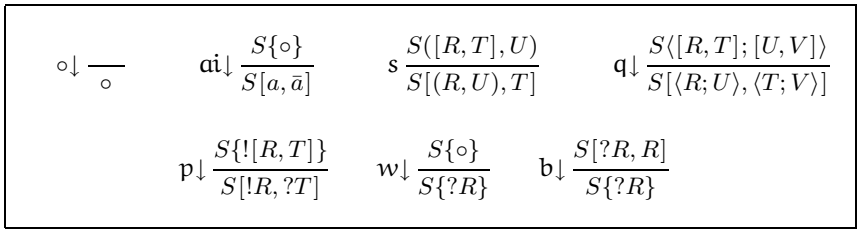

Fig. 2. System NEL

the implication $T \Rightarrow R$ inside a generic context $S\{\ \}$, which is linear implication[4] in our case. An inference rule is called an *axiom* if its premise is empty. Rules with empty contexts correspond to the case of the sequent calculus.

A (formal) *system* $\mathscr{S}$ is a set of inference rules. A derivation Δ in a certain formal system is a finite chain of instances of inference rules in the system. A derivation can consist of just one structure. The topmost structure in a derivation, if present, is called the *premise* of the derivation, and the bottommost structure is called its *conclusion*. A derivation Δ whose premise is T, conclusion is R, and inference rules are in $\mathscr{S}$ will be written as $\begin{array}{c} T \\ \Delta \Vert \mathscr{S} \\ R \end{array}$. Similarly, $\begin{array}{c} \Pi \Vert \mathscr{S} \\ R \end{array}$ will denote a *proof* Π which is a finite derivation whose topmost inference rule is an axiom.

The system in Figure 2 is called *Non-commutative Exponential Linear logic*, or system NEL. The rules of the system are *unit* ($\circ\downarrow$), *atomic interaction* ($\mathsf{ai}\downarrow$), *switch* (s), *seq* ($\mathsf{q}\downarrow$), *promotion* ($\mathsf{p}\downarrow$), *weakening* ($\mathsf{w}\downarrow$), and *absorption* ($\mathsf{b}\downarrow$).

For system NEL, the cut rule has the shape $\mathsf{i}\uparrow \dfrac{S(R,\bar{R})}{S\{\circ\}}$.

Theorem 1 (Cut Elimination). [10] *The rule* $\mathsf{i}\uparrow$ *is admissible for system* NEL*, in other words, for every proof* $\begin{array}{c} \Pi \Vert \mathsf{NEL}\cup\{\mathsf{i}\uparrow\} \\ R \end{array}$ *, there is a proof* $\begin{array}{c} \Pi' \Vert \mathsf{NEL} \\ R \end{array}$ *.*

Theorem 2 (Decomposition). [19] *For every derivation* Δ *in system* NEL*, there is a derivation* Δ' *where, seen bottom-up, first system* $\{\mathsf{b}\downarrow\}$*, then* $\{\mathsf{w}\downarrow\}$*, and then* $\{\mathsf{p}\downarrow, \mathsf{s}, \mathsf{q}\downarrow, \mathsf{ai}\downarrow\}$ *are applied.*

There is a straightforward correspondence between structures not involving seq and formulae of multiplicative exponential linear logic (MELL). For example $![(?a, b), \bar{c}, !\,\bar{d}]$ corresponds to $!\,((?a \otimes b) \parr c^{\perp} \parr !\,d^{\perp})$, and vice versa. Units 1 and

[4] Due to duality between $T \Rightarrow R$ and $\bar{R} \Rightarrow \bar{T}$, rules come in pairs of dual rules: a down-version and an up-version. For instance, the dual of the $\mathsf{ai}\downarrow$ rule is the cut rule. In this paper, we only consider the down rules, since the up rules, including the cut rule, are admissible.

$\bot$ are mapped into $\circ$, since $1 \equiv \bot$, when the rules $\mathtt{mix}$ and $\mathtt{mix0}$ are added to MELL. For a proof of the above results, a more detailed discussion on the proof theory of NEL and the precise relation between NEL and MELL, the reader is referred to [19].

4 Planning with NEL

In this section, we present our encoding of the planning problems in the language of NEL and show that it is correct with respect to conjunctive planning problems.

Definition 1. *The* sequential action structure *for an action,* $a : \{\!|\, c_1, \ldots, c_p \,|\!\} \rightarrow \{\!|\, e_1, \ldots, e_q \,|\!\}$*, denoted by* $\mathcal{Q}$*, is the structure* $\langle [\bar{c}_1, \ldots, \bar{c}_p]; a; [e_1, \ldots, e_q] \rangle$.

Definition 2. *The* simple problem structure P^s *for an initial state* $\mathcal{I} = \{\!|\, r_1, \ldots, r_m \,|\!\}$ *and a goal state* $\mathcal{G} = \{\!|\, g_1, \ldots, g_n \,|\!\}$ *is the structure* $[r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n]$.

Because an action can be executed arbitrarily many times, we employ the exponential "?" which retains a controlled contraction and weakening on the action structures. This way, we can duplicate an action structure by applying the $\mathtt{b}{\downarrow}$ rule, or annihilate it by applying the $w{\downarrow}$ rule during the search for the plans. This also allows us to make the interaction between the planning problems and actions explicit by prefixing a planning problem structure with "!": by applying the $\mathtt{p}{\downarrow}$ rule in proof search we allow an action structure to get inside and interact with a problem structure.

We can now define a planning problem in the language of NEL.

Definition 3. *Let* $\mathscr{P} = \langle \mathcal{I}, \mathcal{G}, \mathscr{A}, \mathcal{F} \rangle$ *be a planning problem. The* sequential conjunctive planning problem structure *(shortly scpps) for* $\mathscr{P}$*, denoted by* $\mathcal{P}^s$*, is the structure*

$$[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, !\, P^s, ?\bar{f}_1, \ldots, ?\bar{f}_h]$$

where $\mathcal{Q}_i$ $(1 \leq i \leq k)$ *are the sequential action structures for the actions in* $\mathscr{A}$*,* P^s *is the simple problem structure for* $\mathcal{I}$ *and* $\mathcal{G}$*, and* $\mathcal{F} = \{f_1, \ldots, f_h\}$.

Let us reconsider the conjunctive planning problem from Section 2. This planning problem can be expressed as the following scpps.

$$[\, ?\langle \bar{d}; c_d; [f, f] \rangle\,,\ ?\langle \bar{f}; b_l; l \rangle\,,\ !\,[d, \bar{l}],\ ?d, ?f, ?l\,] \tag{1}$$

The structures $?\langle \bar{d}; c_d; [f, f] \rangle$ and $?\langle \bar{f}; b_l; l \rangle$ are the sequential action structures for the actions c_d and b_l, respectively. The structure $[d, \bar{l}]$ is the simple problem structure for the initial state $\mathcal{I} = \{\!|\, d \,|\!\}$ and the goal state $\mathcal{G} = \{\!|\, l \,|\!\}$. The structures $?d, ?l$ and $?f$ correspond to $\mathcal{F} = \{d, f, l\}$.

In the following, we will show that searching for a certain kind of derivations where the conclusion is the scpps for a planning problem is equivalent to finding a solution for this planning problem. With the following lemmata, we will formally express the operational semantics of reaching a goal state and applying an action to a state in the language of NEL, respectively.

Lemma 1. *The following rule is derivable in* NEL.

$$\mathit{termination}\ \frac{S\{!\,P\}}{S[!\,\langle P; [r_1, \ldots, r_n, g_1, \ldots, g_s, \bar{r}_1, \ldots, \bar{r}_n]\rangle, ?\bar{f}_1, \ldots, ?\bar{f}_h]}$$

where $g_i \in \{f_1, \ldots, f_h\}$ *for* $1 \leq i \leq s$.

Proof: Take the following derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{S\{!\,P\}}{\vdots}\,\mathrm{ai}\downarrow}{!\,\langle P; [g_1, \ldots, g_s, \bar{g}_1, \ldots, \bar{g}_s]\rangle}\,\mathrm{ai}\downarrow}{!\,[\langle P; [g_1, \ldots, g_s]\rangle, \bar{g}_1, \ldots, \bar{g}_s]}\,\mathrm{q}\downarrow}{\vdots}\,\mathrm{p}\downarrow}{[!\,\langle P; [g_1, \ldots, g_s]\rangle, ?\bar{g}_1, \ldots, ?\bar{g}_s]}\,\mathrm{p}\downarrow}{\vdots}\,w\downarrow}{[!\,\langle P; [g_1, \ldots, g_s]\rangle, ?\bar{g}_1, \ldots, ?\bar{g}_s, ?\bar{f}_1, \ldots, ?\bar{f}_h]}\,w\downarrow}{\vdots}\,\mathrm{b}\downarrow}{[!\,\langle P; [g_1, \ldots, g_s]\rangle, ?\bar{f}_1, \ldots, ?\bar{f}_h]}\,\mathrm{b}\downarrow}{\vdots}\,\mathrm{ai}\downarrow}{[!\,\langle P; [r_1, \ldots, r_n, g_1, \ldots, g_s, \bar{r}_1, \ldots, \bar{r}_n]\rangle, ?\bar{f}_1, \ldots, ?\bar{f}_h]}\,\mathrm{ai}\downarrow$$

□

Lemma 2. *The following rule is derivable in* NEL.

$$\mathit{action}\ \frac{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; a; [R, E]\rangle\,]}{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; [c_1, \ldots, c_p, R]\rangle\,]}$$

Proof: Take the following derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; a; [E, R]\rangle\,]}{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; [\langle a; E\rangle,\ R]\rangle\,]}\,\mathrm{q}\downarrow}{\vdots}\,\mathrm{ai}\downarrow}{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; [\langle[c_1, \ldots, c_p, \bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle,\ R]\rangle\,]}\,\mathrm{ai}\downarrow}{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; [\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle,\ c_1, \ldots, c_p], R\rangle\,]}\,\mathrm{q}\downarrow}{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,[\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,\langle P; [c_1, \ldots, c_p, R]\rangle]\,]}\,\mathrm{q}\downarrow}{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; [c_1, \ldots, c_p, R]\rangle\,]}\,\mathrm{p}\downarrow}{S[\,?\langle[\bar{c}_1, \ldots, \bar{c}_p]; a; E\rangle\,,\,!\,\langle P; [c_1, \ldots, c_p, R]\rangle\,]}\,\mathrm{b}\downarrow$$

□

By employing the rules *action* and *termination* bottom-up, we can search for plans while going up in a derivation: the rule *action* is applied till the multiset of negative atoms in the of-course structure denoting the simple problem structure is a submultiset of the multiset of positive atoms, where the rule *termination* can be applied. After annihilating the of-course structures for the sequential action structures and excessive resources with the rule $w\downarrow$, such a derivation will then give a plan structure at the premise which is a solution for the planning problem. The following theorem proves that our encoding is correct.

Theorem 3. *Let $\mathscr{P} = \langle \mathcal{I}, \mathcal{G}, \mathscr{A}, \mathcal{F} \rangle$ be a conjunctive planning problem and $\mathcal{P}^s$ the scpps for $\mathscr{P}$. There is a derivation*

$$\begin{array}{c} !\,p \\ \| \mathsf{NEL} \\ \mathcal{P}^s \end{array}$$

iff the plan p solves $\mathscr{P}$.

Proof:
(⇒:) Proof with induction on the length of the plan at the premise of the derivation. If there is a derivation with the empty plan $\circ$ at the premise of the derivation, then it follows from Lemma 1 that there must be a derivation of the following form.

$$termination \dfrac{w\downarrow \dfrac{w\downarrow \dfrac{\circ}{\vdots}}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k ?\bar{f}_1, \ldots, ?\bar{f}_{h'}]}}{= \dfrac{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, !\,\langle \circ;\, [r_1, \ldots, r_n, f_1, \ldots, f_s, \bar{r}_1, \ldots, \bar{r}_n] \rangle, ?\bar{f}_1, \ldots, ?\bar{f}_h]}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, !\, [r_1, \ldots, r_n, f_1, \ldots, f_s, \bar{r}_1, \ldots, \bar{r}_n], ?\bar{f}_1, \ldots, ?\bar{f}_h]}} \quad (2)$$

It follows that for $\mathcal{I} \dot{\supseteq} \{\!| r_1, \ldots, r_n |\!\} = \mathcal{G}$, plan $\circ$ with length 0 solves the planning problem. Turning to the induction step, we assume that if there is a derivation

$$\begin{array}{c} \langle a_1; \ldots; a_j \rangle \\ \Delta \| \mathsf{NEL} \\ [?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, !\, [r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n], ?\bar{f}_1, \ldots, ?\bar{f}_h] \end{array} , \qquad (3)$$

then the plan $\langle\, a_1; \ldots; a_j \,\rangle$ solves the planning problem $\langle \mathcal{I}, \mathcal{G}, \mathscr{A}, \mathcal{F} \rangle$ where $\mathscr{A} = \{a_1, \ldots, a_k\}$, $\mathcal{I} = \{\!| r_1, \ldots, r_m |\!\}$ and $\mathcal{G} = \{\!| g_1, \ldots, g_n |\!\}$. If there is such a derivation Δ, then it follows from Theorem 2 that there is a derivation of the following form.

$$
\begin{array}{c}
\langle\, a_1; \ldots; a_j \,\rangle \\
{\scriptstyle \Delta_1 \,[\!|\, \{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\}} \\
!\,[\mathcal{Q}_1, \ldots, \mathcal{Q}_j, r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n, \bar{f}_1, \ldots, \bar{f}_{h''}] \\
{\scriptstyle \Delta_2 \,[\!|\, \{\mathtt{p}\downarrow\}} \\
[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_j, !\,[r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n], ?\bar{f}_1, \ldots, ?\bar{f}_{h''}] \\
{\scriptstyle \Delta_3 \,[\!|\, \{w\downarrow\}} \\
[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, ?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_j, !\,[r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n], ?\bar{f}_1, \ldots, ?\bar{f}_{h''}, ?\bar{f}_1, \ldots, ?\bar{f}_h] \\
{\scriptstyle \Delta_4 \,[\!|\, \{\mathtt{b}\downarrow\}} \\
[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, !\,[r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n], ?\bar{f}_1, \ldots, ?\bar{f}_h]
\end{array}
$$

Consider the planning problem given with $\mathcal{A}$, $\mathcal{I}' = \{\!|\, c_1, \ldots, c_p, r_{q+1}, \ldots, r_m \,|\!\}\ \mathcal{G}$ and $\mathcal{F}'$ such that $\mathcal{F} \subseteq \{f'_1, \ldots, f'_{h'}\} = \mathcal{F}'$. Suppose there is a derivation with a plan of length $j+1$ at the premise. Then, from Lemma 2 there must exist a derivation of the following form.

$$
\begin{array}{c}
!\,\langle a; a_1; \ldots; a_j \rangle \\
{\scriptstyle \Delta_1 \,[\!|\, \{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\}} \\
\mathtt{q}\downarrow \dfrac{!\,\langle a; [\mathcal{Q}_1, \ldots, \mathcal{Q}_j, r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n, \bar{f}_1, \ldots, \bar{f}_{h''}] \rangle}{!\,[\mathcal{Q}_1, \ldots, \mathcal{Q}_j, \langle a; [r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \rangle, \bar{f}_1, \ldots, \bar{f}_{h''}]} \\
\mathtt{p}\downarrow \dfrac{}{} \\
\vdots \\
\mathtt{p}\downarrow \dfrac{}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_j, !\,\langle a; [r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \rangle, ?\bar{f}_1, \ldots, ?\bar{f}_{h''}]} \\
w\downarrow \dfrac{}{} \\
\vdots \\
w\downarrow \dfrac{}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, ?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_j, !\,\langle a; [r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \rangle, ?\bar{f}_1, \ldots, ?\bar{f}_h, ?\bar{f}_1, \ldots, ?\bar{f}_{h''}]} \\
\mathtt{b}\downarrow \dfrac{}{} \\
\vdots \\
\mathtt{b}\downarrow \dfrac{}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, !\,\langle a; [r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \rangle, ?\bar{f}_1, \ldots, ?\bar{f}_h]} \\
w\downarrow \dfrac{}{} \\
\vdots \\
w\downarrow \dfrac{}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, !\,\langle a; [r_1, \ldots, r_q, r_{q+1}, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \rangle, ?\bar{f}'_1, \ldots, ?\bar{f}'_{h'}]} \\
\mathit{action} \dfrac{}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, ?\langle [\bar{c}_1, \ldots, \bar{c}_p]; a; [r_1, \ldots, r_q] \rangle, !\,\langle \circ\,; [c_1, \ldots, c_p, r_{q+1}, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \rangle, ?\bar{f}'_1, \ldots, ?\bar{f}'_{h'}]} \\
= \dfrac{}{} \\
\mathtt{b}\downarrow \dfrac{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_k, ?\langle [\bar{c}_1, \ldots, \bar{c}_p]; a; [r_1, \ldots, r_q] \rangle, !\,[c_1, \ldots, c_p, r_{q+1}, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n], ?\bar{f}'_1, \ldots, ?\bar{f}'_{h'}]}{[?\mathcal{Q}_1, \ldots, ?\mathcal{Q}_i, \ldots, ?\mathcal{Q}_k, !\,[c_1, \ldots, c_p, r_{q+1}, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n], ?\bar{f}'_1, \ldots, ?\bar{f}'_{h'}]}
\end{array}
\tag{4}
$$

Observe that there is an action $a : \{\!|\, c_1, \ldots, c_p \,|\!\} \to \{\!|\, r_1, \ldots, r_q \,|\!\}$. Since

$$
\mathcal{I}' \dot{-} \{\!|\, c_1, \ldots, c_p \,|\!\} \dot{\cup} \{\!|\, r_1, \ldots, r_q \,|\!\} = \{\!|\, r_1, \ldots, r_m \,|\!\},
$$

it follows that $\langle a; a_1; \ldots; a_k \rangle$ with length $k+1$ is a solution for the planning problem.

($\Leftarrow$:) Proof with induction on the length of the plan p. If p is the empty plan, then it must be the case that $\mathcal{I} = \{\!|\, r_1, \ldots, r_m \,|\!\} \dot{\supseteq} \mathcal{G}$. Take the derivation (2). Turning to the induction step we assume that the result holds for a plan with k number of actions. Suppose there is a plan $\langle a; a_1; \ldots; a_k \rangle$ that solves the planning problem given with $\mathcal{I} = \{\!|\, r_1, \ldots, r_m, c_1, \ldots, c_p \,|\!\}$ and $\mathcal{G} = \{\!|\, g_1, \ldots, g_n \,|\!\}$. Hence we find an action

$$a \,:\, \{\!|\, c_1, \ldots, c_p \,|\!\} \rightarrow \{\!|\, r_1, \ldots, r_q \,|\!\} \text{ , where } \; q < m \,,$$

and a planning problem given with

$$\mathcal{I}' = \{\!|\, r_1, \ldots, r_q, r_{q+1}, \ldots, r_m \,|\!\} \,, \quad \mathcal{G} = \{\!|\, g_1, \ldots, g_n \,|\!\}$$

which is solved by $\langle a_1; \ldots; a_k \rangle$. Since we have the derivation Δ in (3) from the induction hypothesis, the derivation (4) proves the result. □

To illustrate the above ideas, let us return to our running example. Observe that the conclusion of the below derivation is the scpps in (1).

$$\dfrac{\dfrac{\dfrac{\dfrac{w\!\downarrow^4 \dfrac{!\,\langle c_d\,;b_l\rangle}{[\,?\langle \bar{d};c_d;[f,f]\rangle\,,\,?\langle \bar{f};b_l;l\rangle\,,\,!\,\langle c_d\,;b_l\rangle,\;?d,?l\,]}}{[\,?\langle \bar{d};c_d;[f,f]\rangle\,,\,?\langle \bar{f};b_l;l\rangle\,,\,!\,\langle c_d\,;b_l\,;[f,l,\bar{l}]\rangle,\;?d,?f,?l\,]}\,\textit{termination}}{[\,?\langle \bar{d};c_d;[f,f]\rangle\,,\,?\langle \bar{f};b_l;l\rangle\,,\,!\,\langle c_d\,;[f,f,\bar{l}]\rangle,\;?d,?f,?l\,]}\,\textit{action}}{[\,?\langle \bar{d};c_d;[f,f]\rangle\,,\,?\langle \bar{f};b_l;l\rangle\,,\,!\,\langle \circ\,;[d,\bar{l}]\rangle,\;?d,?f,?l\,]}\,\textit{action}}{[\,?\langle \bar{d};c_d;[f,f]\rangle\,,\,?\langle \bar{f};b_l;l\rangle\,,\,!\,[d,\bar{l}],\;?d,?f,?l\,]}\,=$$

The plan structure at the premise is a solution of our planning problem.

5 Concurrent Plans

As well as sequential composition due to non-commutative *seq* operator, the language of NEL allows to express parallel composition of plans and actions by employing the commutative *par* operator. In this section, we further extend the notion of plans to the notion of concurrent plans, and show that our encoding of the planning problems allows to capture parallelism in plans.

Definition 4. *A* concurrent plan structure *is a structure generated by*

$$P ::= \circ \mid a \mid \langle\, P\,;P\,\rangle \mid [\,P\,,P\,]$$

where a denotes atoms representing actions.

Proposition 1. *For every planning problem $\mathscr{P}$ given with $\mathcal{I}$, $\mathcal{G}$, and $\mathscr{A}$, and a plan $\langle a_1; \ldots; a_k \rangle$ that solves it, for some $s \leq k$, there is a planning problem $\mathscr{P}'$ given with $\mathcal{I}' = \Phi(a_s, \ldots, \Phi(a_1, \mathcal{I}) \ldots)$, $\mathcal{G}$ and $\mathscr{A}$ that is solved by $\langle a_{s+1}; \ldots; a_k \rangle$.*

Proof: Follows immediately from the definitions in Section 2. □

Proposition 2. *Let $\mathcal{I}$, $\mathcal{S}_1$, $\mathcal{S}_2$ be states and p be a plan.*
$\Phi(p,\mathcal{I}) = \mathcal{S}_1$ *iff* $\Phi(p\,,\,\mathcal{I}\,\dot\cup\,\mathcal{S}_2) = \mathcal{S}_1\,\dot\cup\,\mathcal{S}_2\,.$

Proof: With induction on the length of p. □

Lemma 3. *Let $\mathcal{I}_1 = \{\!|\, r_1,\ldots,r_m \,|\!\}$, $\mathcal{I}_2 = \{\!|\, r'_1,\ldots,r'_n \,|\!\}$, $\mathcal{S}_1 = \{\!|\, g_1,\ldots,g_p \,|\!\}$ and $\mathcal{S}_2 = \{\!|\, g'_1,\ldots,g'_q \,|\!\}$ be states and $p_1 = \langle a_1,\ldots,a_k\rangle$, $p_2 = \langle a'_1,\ldots,a'_{k'}\rangle$ be plans. Furthermore, let $\mathcal{Q}_1,\ldots,\mathcal{Q}_k,\mathcal{Q}'_1,\ldots,\mathcal{Q}'_{k'}$ be the sequential action structures for the actions $a_1,\ldots,a_k,a'_1,\ldots,a'_{k'}$. The following are equivalent.*

(i) $\Phi(\,p_1,\Phi(\,p_2,\,\mathcal{I}_1\,\dot\cup\,\mathcal{I}_2\,)) = \Phi(\,p_2,\Phi(\,p_1,\mathcal{I}_1\,\dot\cup\,\mathcal{I}_2\,)) = \mathcal{S}\,.$

(ii) $\Phi(p_1,\mathcal{I}_1) = \mathcal{S}_1$ *and* $\Phi(p_2,\mathcal{I}_2) = \mathcal{S}_2$ *such that* $\mathcal{S} = \mathcal{S}_1\,\dot\cup\,\mathcal{S}_2\,.$

(iii)

$$\begin{array}{c}\langle[p_1,p_2];[g_1,\ldots,g_p,g'_1,\ldots,g'_q]\rangle\\ \scriptstyle{[\![\,\{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\}}\\ [\mathcal{Q}_1,\ldots,\mathcal{Q}_k,\mathcal{Q}'_1,\ldots,\mathcal{Q}'_{k'},r_1,\ldots,r_m,r'_1,\ldots,r'_n]\end{array}$$

Proof:
(i) ⇒ *(ii)* : Let $\Phi(p_1,\mathcal{I}_1) = \mathcal{S}'$. From Proposition 2, we have
$\Phi(p_2\,,\,\Phi(p_1\,,\,\mathcal{I}_1\,\dot\cup\mathcal{I}_2)) = \Phi(p_2\,,\,\mathcal{S}'\,\dot\cup\mathcal{I}_2) = \mathcal{S}\,.$
Assume that *(i)* holds and *(ii)* does not hold.
This can only be the case when there are fluents in $\mathcal{S}'$ that are not present in $\mathcal{I}_2$ and consumed by p_2, but this contradicts with $\Phi(\,p_2,\Phi(\,p_1,\mathcal{I}_1\,\dot\cup\,\mathcal{I}_2\,)) = \mathcal{S}$

(ii) ⇒ *(iii)* : Observe that *(ii)* implies that there are the following derivations.

$$\begin{array}{c}\langle p_1;[g_1,\ldots,g_p]\rangle\\ \scriptstyle{\Delta_1\,[\![\,\{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\}}\\ [\mathcal{Q}_1,\ldots,\mathcal{Q}_k,r_1,\ldots,r_m]\end{array}\qquad\qquad\begin{array}{c}\langle p_2;[g'_1,\ldots,g'_q]\rangle\\ \scriptstyle{\Delta_2\,[\![\,\{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\}}\\ [\mathcal{Q}'_1,\ldots,\mathcal{Q}'_{k'},r'_1,\ldots,r'_n]\end{array}$$

Take the following derivation.

$$\begin{array}{c}\mathtt{q}\downarrow\dfrac{\langle[p_1,p_2];[g_1,\ldots,g_p,g'_1,\ldots,g'_q]\rangle}{[\langle p_1;[g_1,\ldots,g_p]\rangle,\langle p_2;[g'_1,\ldots,g'_q]\rangle]}\\ \scriptstyle{\Delta_1\,[\!|}\\ [\mathcal{Q}_1,\ldots,\mathcal{Q}_k,r_1,\ldots,r_m,\langle p_2;[g'_1,\ldots,g'_q]\rangle]\\ \scriptstyle{\Delta_2\,[\!|}\\ [\mathcal{Q}_1,\ldots,\mathcal{Q}_k,\mathcal{Q}'_1,\ldots,\mathcal{Q}'_{k'},r_1,\ldots,r_m,r'_1,\ldots,r'_n]\end{array}$$

$(iii) \Rightarrow (i)$: The following derivations together with Theorem 3 prove the result.

$$\begin{array}{c} = \dfrac{\langle p_1; p_2\rangle}{\langle [p_1, \circ]; [\circ, p_2]\rangle} \\ \mathfrak{q}\!\downarrow \dfrac{}{[\langle p_1; \circ\rangle, \langle \circ; p_2\rangle]} \\ = \dfrac{}{[p_1, p_2]} \end{array} \qquad \begin{array}{c} = \dfrac{\langle p_2; p_1\rangle}{\langle [\circ, p_2]; [p_1, \circ]\rangle} \\ \mathfrak{q}\!\downarrow \dfrac{}{[\langle \circ; p_1\rangle, \langle p_2; \circ\rangle]} \\ = \dfrac{}{[p_1, p_2]} \end{array}$$

□

Definition 5. *A concurrent plan structure* P solves a planning problem $\mathscr{P}$*, if, for all the derivations*

$$\begin{array}{c} p \\ \|_{\{\mathfrak{q}\downarrow\}} \\ P \end{array}$$

where p *is a plan structure,* p *solves* $\mathscr{P}$*.*

To illustrate these ideas let us return to our running example. However, this time Bert is not only thirsty but also hungry. Since he is equipped with the action that allows him to *get a candy-bar* (c) for 50 cents from the vending machine, this should not be a problem. Then, once he has a lemonade and a candy bar, he can *have lunch* which makes him *happy* (h). Consider the following scpps

$$[\,?\langle \bar{d}; c_d; [f, f]\rangle\,,\; ?\langle \bar{f}; b_l; l\rangle\,,\; ?\langle \bar{f}; g_c; c\rangle\,,\; ?\langle [\bar{l}, \bar{c}]; h_l; h\rangle\,,\; !\,[d, \bar{h}],\; ?d, ?f, ?l, ?c, ?h\,]$$

where $?\langle \bar{f}; g_c; c\rangle$ and $?\langle [\bar{l}, \bar{c}]; h_l; h\rangle$, respectively, are the sequential action structures for the actions *get a candy-bar* and *have lunch*, respectively. It is easy to observe that the concurrent plan structure

$$\langle c_d\,; [b_l\,, g_c]; h_l\rangle$$

solves the above planning problem. The following theorem formally justifies that there is a derivation which provides this concurrent plan structure at the premise.

Theorem 4. *Let* $\mathscr{P}$ *be a planning problem and* $\mathcal{P}^s$ *be the scpps for* $\mathscr{P}$*. If* P *is a concurrent plan structure that solves a planning problem* $\mathscr{P}$*, then there is a derivation of the following form.*

$$\begin{array}{c} !\,P \\ \Delta\|_{\mathsf{NEL}} \\ \mathcal{P}^s \end{array}$$

Proof: Let $p = \langle a_1, \ldots, a_k\rangle$ be a plan structure such that there is a derivation $\begin{array}{c} p \\ \|_{\{\mathfrak{q}\downarrow\}} \\ P \end{array}$. From Theorem 3 there is a derivation $\begin{array}{c} !\,\langle a_1, \ldots, a_k\rangle \\ \|_{\mathsf{NEL}} \\ \mathcal{P}^s \end{array}$ and from Theorem

2, there is a derivation of the following form

$$\begin{array}{c} !\,\langle a_1,\ldots,a_k\rangle \\ \Delta_1 \,[\!|\, \{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\} \\ !\,[\mathcal{Q}_1,\ldots,\mathcal{Q}_k,r_1,\ldots,r_m,\bar{g}_1,\ldots,\bar{g}_n] \\ \Delta_2 \,[\!|\, \{\mathtt{p}\downarrow,\mathtt{w}\downarrow,\mathtt{b}\downarrow\} \\ \mathcal{P}^s \end{array}$$

where $\mathcal{Q}_1,\ldots,\mathcal{Q}_k$ are the sequential action structures for the actions $a_1,\ldots,a_k$. It remains to prove that there is a derivation

$$\begin{array}{c} P \\ \Delta_3 \,[\!|\, \{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\} \\ [\mathcal{Q}_1,\ldots,\mathcal{Q}_k,r_1,\ldots,r_m,\bar{g}_1,\ldots,\bar{g}_n] \end{array}$$

We will construct the derivation Δ_3 with structural induction on P.

– If $P = \circ$ or $P = a$, then take Δ_1.

– If $P = \langle P_1; P_2\rangle$, then there must be a plan $p = \langle p_1; p_2\rangle$ that solves $\mathscr{P}$ such that

$$\begin{array}{c} p_1 \\ [\!|\, \{\mathtt{q}\downarrow\} \\ P_1 \end{array} \quad \text{and} \quad \begin{array}{c} p_2 \\ [\!|\, \{\mathtt{q}\downarrow\} \\ P_2 \end{array}$$

where $p_1 = \langle a_1,\ldots,a_{k'}\rangle$ and $p_2 = \langle a_{k'+1},\ldots,a_k\rangle$.
Let $\mathcal{Q}_1,\ldots,\mathcal{Q}_k$ be the sequential action structures for the actions $a_1,\ldots,a_k$. Then from Proposition 1 and Theorem 3, there must be a derivation of the following form.

$$\begin{array}{c} \mathtt{ai}\downarrow \dfrac{\langle p_1; p_2\rangle}{\vdots} \\ \mathtt{ai}\downarrow \dfrac{}{\langle p_1; [t_1,\ldots,t_h,\bar{t}_1,\ldots,\bar{t}_h]; p_2\rangle} \\ \mathtt{q}\downarrow \dfrac{}{\langle p_1; [t_1,\ldots,t_h,\langle[\bar{t}_1,\ldots,\bar{t}_h]; p_2\rangle]\rangle} \\ \mathtt{q}\downarrow \dfrac{}{[\langle p_1; [t_1,\ldots,t_h]\rangle,\langle[\bar{t}_1,\ldots,\bar{t}_h]; p_2\rangle]} \\ \Big\|\, \{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\} \\ [\mathcal{Q}_1,\ldots,\mathcal{Q}_{k'},r_1,\ldots,r_{m'},\bar{g}_1,\ldots,\bar{g}_{n'},\langle[\bar{t}_1,\ldots,\bar{t}_h]; p_2\rangle] \\ [\!|\, \{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\} \\ = \dfrac{[\mathcal{Q}_1,\ldots,\mathcal{Q}_{k'},r_1,\ldots,r_{m'},\bar{g}_1,\ldots,\bar{g}_{n'},\mathcal{Q}_{k'+1},\ldots,\mathcal{Q}_k,r_{m'+1},\ldots,r_m,\bar{g}_{n'+1},\ldots,\bar{g}_n]}{[\mathcal{Q}_1,\ldots,\mathcal{Q}_k,r_1,\ldots,r_m,\bar{g}_1,\ldots,\bar{g}_n]} \end{array}$$

It follows that induction hypothesis gives the derivations

$$\begin{array}{c} \langle P_1; [t_1,\ldots,t_h]\rangle \\ \Delta_4 \,[\!|\, \{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\} \\ [\mathcal{Q}_1,\ldots,\mathcal{Q}_{k'},r_1,\ldots,r_{m'},\bar{g}_1,\ldots,\bar{g}_{n'}] \end{array} \quad \text{and} \quad \begin{array}{c} \langle[\bar{t}_1,\ldots,\bar{t}_h]; P_2\rangle \\ \Delta_5 \,[\!|\, \{\mathtt{ai}\downarrow,\mathtt{q}\downarrow\} \\ [\mathcal{Q}_{k'+1},\ldots,\mathcal{Q}_k,r_{m'+1},\ldots,r_m,\bar{g}_{n'+1},\ldots,\bar{g}_n] \end{array}.$$

Take the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\langle P_1; P_2\rangle}{\vdots}\,\mathtt{ai}\downarrow}{\langle P_1; [t_1, \ldots, t_h, \bar{t}_1, \ldots, \bar{t}_h]; P_2\rangle}\,\mathtt{ai}\downarrow}{\langle P_1; [t_1, \ldots, t_h, \langle [\bar{t}_1, \ldots, \bar{t}_h]; P_2\rangle]\rangle}\,\mathtt{q}\downarrow}{[\langle P_1; [t_1, \ldots, t_h]\rangle, \langle [\bar{t}_1, \ldots, \bar{t}_h]; P_2\rangle]}\,\mathtt{q}\downarrow}$$

$$\begin{array}{c} [\Delta_4, \Delta_5]\,\big\|\{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\} \\ [\mathcal{Q}_1, \ldots, \mathcal{Q}_k, r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \end{array} .$$

– If $P = [P_1, P_2]$ then there must be plans $p = \langle p_1; p_2\rangle$ and $p' = \langle p_2; p_1\rangle$ that solve $\mathscr{P}$ such that

$$\begin{array}{c} p_1 \\ \big\|\{\mathtt{q}\downarrow\} \\ P_1 \end{array} \quad \text{and} \quad \begin{array}{c} p_2 \\ \big\|\{\mathtt{q}\downarrow\} \\ P_2 \end{array} .$$

From Lemma 3 and Theorem 3 there must be a derivation of the following form.

$$\begin{array}{c} [p_1, p_2] \\ \big\|\{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\} \\ [\mathcal{Q}_1, \ldots, \mathcal{Q}_{k'}, r_1, \ldots, r_{m'}, \bar{g}_1, \ldots, \bar{g}_{n'}, p_2] \\ \big\|\{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\} \\ [\mathcal{Q}_1, \ldots, \mathcal{Q}_{k'}, r_1, \ldots, r_{m'}, \bar{g}_1, \ldots, \bar{g}_{n'}, \mathcal{Q}_{k'+1}, \ldots, \mathcal{Q}_k, r_{m'+1}, \ldots, r_m, \bar{g}_{n'+1}, \ldots, \bar{g}_n] \end{array} = \dfrac{[\mathcal{Q}_1, \ldots, \mathcal{Q}_{k'}, r_1, \ldots, r_{m'}, \bar{g}_1, \ldots, \bar{g}_{n'}, \mathcal{Q}_{k'+1}, \ldots, \mathcal{Q}_k, r_{m'+1}, \ldots, r_m, \bar{g}_{n'+1}, \ldots, \bar{g}_n]}{[\mathcal{Q}_1, \ldots, \mathcal{Q}_k, r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n]}$$

It follows that induction hypothesis gives the derivations

$$\begin{array}{c} P_1 \\ \Delta_6\,\big\|\{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\} \\ [\mathcal{Q}_1, \ldots, \mathcal{Q}_{k'}, r_1, \ldots, r_{m'}, \bar{g}_1, \ldots, \bar{g}_{n'}] \end{array} \quad \text{and} \quad \begin{array}{c} P_2 \\ \Delta_7\,\big\|\{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\} \\ [\mathcal{Q}_{k'+1}, \ldots, \mathcal{Q}_k, r_{m'+1}, \ldots, r_m, \bar{g}_{n'+1}, \ldots, \bar{g}_n] \end{array} .$$

Take the derivation

$$\begin{array}{c} [P_1, P_2] \\ [\Delta_6, \Delta_7]\,\big\|\{\mathtt{ai}\downarrow, \mathtt{q}\downarrow\} \\ [\mathcal{Q}_1, \ldots, \mathcal{Q}_k, r_1, \ldots, r_m, \bar{g}_1, \ldots, \bar{g}_n] \end{array} .$$

□

Corollary 1. *Let $\mathscr{P}$ be a planning problem and $\mathcal{P}^s$ be the scpps for $\mathscr{P}$. If P is a concurrent plan structure that solves a planning problem $\mathscr{P}$, then the structure $[\mathcal{P}^s, ?\bar{P}]$ has a proof in* NEL.

Proof: Result follows immediately from Theorem 4.

6 Logical Strategies

We did not make any use of the rule s in our derivations. However, this rule can be employed to logically enforce strategies for backward or forward search for plans, where the search starts from the inital state or the goal state. For instance, consider the following schemes of derivations with the sequential action structures and simple problem structures at the premise.

$$\mathsf{s}\,\dfrac{\langle(\bar{c}_1,\ldots,\bar{c}_p);a;[e_1,\ldots,e_q]\rangle}{\vdots}\qquad\text{and}\qquad\mathsf{s}\,\dfrac{[r_1,\ldots,r_m,(\bar{g}_1,\ldots,\bar{g}_n)]}{\vdots}$$
$$\mathsf{s}\,\dfrac{}{\langle[\bar{c}_1,\ldots,\bar{c}_p];a;[e_1,\ldots,e_q]\rangle}\qquad\qquad\mathsf{s}\,\dfrac{}{[r_1,\ldots,r_m,\bar{g}_1,\ldots,\bar{g}_n]}$$

By applying these schemes to all the sequential action structures and to the simple planning problem structure in a scpps the search for plans is forced to be forward search since only the atoms in a par structure can interact with the negative atoms in the condition of an action. In order to observe this, the reader can try to apply the induction argument of the proof of Theorem 3 backward by starting from the goal state and applying the last action: this does not work due to possible atoms in the goal state that are not necessarily produced by the last action of the plan.

The same method can be analogously applied for enforcing a backward search strategy.

7 Implementation in Maude

Besides other properties, one interesting and important property which distinguishes calculus of structures from the sequent calculus is deep inference: the calculus of structures does not depend on a notion of main connective, and like in term rewriting, inference rules are deeply applicable inside expressions. This makes it possible to express logical systems as term rewriting systems modulo equality [11, 14].

The language Maude [3] allows implementing term rewriting systems modulo equational theories due to the built in very fast matching algorithm that supports different combinations of associative, commutative equational theories, also with the presence of units. Another important feature of Maude that makes it a plausible platform for implementing systems of the calculus of structures is the availability of the search function since the 2.0 release of Maude. This function implements breadth-first search which is vital for complete search for derivations and proofs.

Exploiting these features, we implemented system NEL in Maude 2 within two modules: the first module converts an arbitrary NEL structure into a structure in negation normal form, i.e., to a structure where there are no occurrences of units and where negation is pushed to the atoms. The second module implements the inference rules of the system modulo associativity, commutativity

and equalities for unit(s). Different modules for the systems in the calculus of structures including NEL are available for download at
`http://www.informatik.uni-leipzig.de/~ozan/maude_cos.html/`.

8 Discussion

We presented an encoding of the conjunctive planning problems in the language of NEL where plans are not extracted from the proof of a derivation, but explicit premises of derivations. Furthermore, we showed that our encoding is expressive enough to capture concurrent plans where plans respecting parallelism can also be obtained.

A direct consequence of this work is the establishment of a bridge between concurrency theory and planning, which allows to apply methods from concurrency to planning: labeled event structures is a model for concurrency [18] which has been studied for linear logic proofs in the sequent calculus proofs in [8]. Future work includes investigating a labeled event structure semantics in our derivations which will result in a notion of plan equivalence.

Our results show that NEL can serve as an operational semantics for conjunctive planning problems. In conjunction with our implementation of system NEL, our result is, with future development, also of practical interest. We conjecture that the rules *action* and *termination* can be expressed in Maude as conditional rewrite rules due to the matching of atoms with opposite polarities. However, such an implementation will disregard the concurrent plans. Because system NEL is undecidable [20] an implementation for practical purposes that captures concurrent plans will require introduction of strategies at the Maude meta-level [4].

References

1. Wolfgang Bibel. A deductive solution for plan generation. In *New Generation Computing*, pages 115–132. 1986.
2. Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming, 18th International Conference*, volume 2401 of *Lecture Notes in Computer Science*, pages 302–316. Springer-Verlag, 2002.
3. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. The Maude 2.0 system. In Robert Nieuwenhuis, editor, *Rewriting Techniques and Applications, Proceedings of the 14th International Conference,*, volume 2706. Springer, 2003.
4. M. Clavel, F. Durán, S. Eker, J. Meseguer, and M.-O. Stehr. Maude as a formal meta-tool. In *World Congress on Formal Methods (2)*, pages 1684–1703, 1999.
5. S. Cresswell, A. Smaill, and J. Richardson. Deductive synthesis of recursive plans in linear logic. In *ECP*, 1999.
6. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
7. G. Große, S. Hölldobler, and J. Schneeberger. Linear deductive planning. In *Journal of Logic and Computation*, volume 6 (2), pages 233–262. 1996.

8. Alessio Guglielmi. *Abstract Logic Programming in Linear Logic Independence and Causality in a First Order Calculus*. PhD thesis, Universita di Pisa – Genova, 1996.
9. Alessio Guglielmi. A system of interaction and structure. Technical Report WV-02-10, TU Dresden, 2002. to appear in ACM Transactions on Computational Logic.
10. Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In M. Baaz and A. Voronkov, editors, *LPAR 2002*, volume 2514 of *Lecture Notes in Artificial Intelligence*, pages 231–246. Springer-Verlag, 2002.
11. Steffen Hölldobler and Ozan Kahramanoğulları. From the calculus of structures to term rewriting systems. Technical Report WV-04-03, TU Dresden, 2004.
12. S. Hölldobler and J. Schneeberger. A new deductive approach to planning. In *New Generation Computing*, pages 225–244. 1990.
13. Éric Jacopin. Classical AI planning as theorem proving: The case of a fragment of linear logic. In *AAAI Fall Symposium on Automated Deduction in Nonstandard Logics*, pages 62–66, Palo Alto, California, 1993. AAAI Press Publications.
14. Ozan Kahramanoğulları. Implementing system BV of the calculus of structures in Maude 2. Technical report, TU Dresden, 2004. to appear in the proceedings of the ESSLLI-2004 Student Session, Université Henri Poincaré, Nancy, France.
15. N. Kobayashi and A. Yonezawa. Reasoning on actions and change in linear logic programming. Technical report, Department of Information Science, University of Tokyo, 1993.
16. M. Masseron, C. Tollu, and J. Vauzeilles. Generating plans in linear logic. In *Foundations of Software Technology and Theoretical Computer Science*, volume 472 of *Lecture Notes in Computer Science*, pages 63–75. Springer-Verlag, 1990.
17. Robin Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.
18. Vladimiro Sassone, Morgens Nielsen, and Glynn Winskel. Models for concurrency: Towards a classification. In *Theoretical Computer Science*, volume 170 (1–2), pages 297–348. 1996.
19. Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, TU Dresden, 2003.
20. Lutz Straßburger. System NEL is undecidable. In Ruy De Queiroz, Elaine Pimentel, and Lucília Figueiredo, editors, *10th Workshop on Logic, Language, Information and Computation (WoLLIC)*, volume 84 of *Electronic Notes in Theoretical Computer Science*, 2003.
21. Alwen Fernanto Tiu. Properties of a logical system in the calculus of structures. Technical Report WV-01-06, Technische Universität Dresden, 2001.