A Framework of Controlled Robot Language for Instruction Understanding and Robot Planning

Dang Tran, Yimesker Yihun, Jindong Tan, and Hongsheng He*

***Abstract*— Effective and efficient communication is critical for human-robot collaboration and human-agent teaming. This paper designs a Controlled Robot Language (CRL) and its formal grammar for instruction interpretation and automated robot planning. The CRL framework defines a formal language domain that deterministically maps linguistic commands to logical semantic expressions. As compared to controlled natural language, whose purpose is for general knowledge representation, CRL expressions are particularly designed to parse human instructions, represent contextual knowledge, and automated robot planning. The grammar of CRL is developed in accordance with the IEEE CORA ontology, which defines the majority of formal English domain, accepting large range of intuitive instructions. For sentences outside this domain, CRL checker is used to detect a linguistic patterns, which can be further processed by CRL translator to recover back an equivalent expression in CRL grammar. The final output is formal semantic representation in first-order logic and formal action representation in linear temporal logic. The CRL framework was evaluated on general purpose instruction corpus, demonstrating outperforming in expressiveness while maintaining the certainty property. The effectiveness of the CRL framework was also demonstrated by automated assembly of an IKEA table following natural-language instructions.**

I. Introduction

Reliable communication between humans and intelligent robots is a critical need especially for human-robot collaboration, human-agent teaming, and multiple agent coordination. For most robotic applications, reliable human-robot communication will significantly reduce the chances of unpredictable catastrophes and fatal damages. In addition to physical interaction, natural language has the potential to become the main communication channel for instructing robots, representing contextual knowledge, and providing feedback. From a psychological perspective, *trust* is the grant obstacle preventing human and robot communicate effectively [1]. A robot with dynamic consciousness and optimized precision does not necessarily gain trust from its users. We believe that the lack of reliable natural-language communication is one of the main factors that hinder the advancement of human-agent teaming.

Significant research progress has been made to address the important and challenging problem of reliable natural-language communication in robotic domain [2]–[5]. Robots are deemed to understand natural language if the robot can either (i) extract correct information or (ii) have a logical semantic representation for the context. From the former perspective, natural language understanding is considered as parsing low-level knowledge for action control. The most common approaches in this branch include probabilistic models [6]–[9] and neural-network methods [10], which have demonstrated robust performance in detecting linguistic patterns and representing low-level knowledge. These approaches depend strongly on training data such that low-level knowledge may not semantically equivalent to the original expression. The latter perspective parses the meaning of natural language in a richer way, where the semantics of natural-language expressions are represented in logics [2], [11], [12].

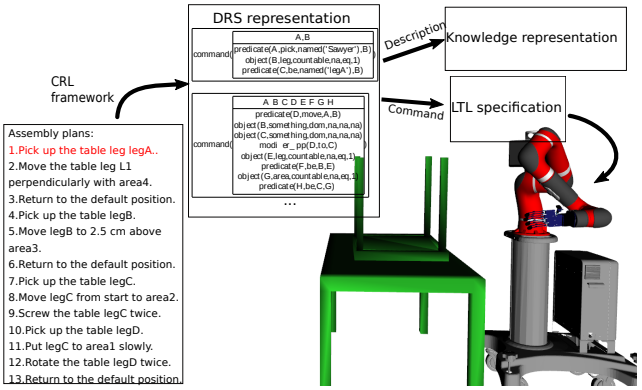

Fig. 1. Parsing of natural-language instructions using CRL. The linguistic instructions from user are converted into appropriate semantic representations represented by Discourse Representation Structure with variables and explicit logical statements. Command-type DRS expression can be used to control robot through LTL specification.

Despite the fruitful research progress of human-robot communication, the appearance of natural language interface in robotics is still limited due to the trade-off between *expressiveness* and *reliability*. Linguistic models can handle a large range of natural language expressions, but they are less deterministic or reliable; on the other hand, systems that can handle natural language more precisely are limited in expressiveness. A common language domain could bridge the gap between expressiveness and reliability in human-robot interaction.

In this paper, we propose a *grammar model* named Controlled Robot Language (CRL) that interprets general human instructions into discourse representation structures

Dang Tran and Hongsheng He are with Department of Electrical Engineer and Computer Science, Wichita State University, Wichita, KS, 67260, USA

Yimesker Yihun is with Department of Mechanical Engineering,, Wichita State University, Wichita, KS, 67260, USA

Jindong Tan is with Department of Mechanical, Aerospace, and Biomedical Engineering, University of Tennessee, Knoxville, TN, USA

*Correspondence should be addressed to Hongsheng He, hongsheng.he@wichita.edu.

(DRS) [13], which is a semantic representation that can capture long-term dependency. More importantly, the CRL is designed as a general-purpose grammar that represents a *deterministic* and *expressive* linguistic domain rather than ad-hoc development, inspired by controlled natural language (CNL) [14]. The CRL framework defines the CRL grammar that syntactically analyzes a majority of English expressions. For dialect expressions that do not strictly follow the grammar, we developed a CRL checker, which contains flexible set of common linguistic patterns, to detect correctable grammar errors and automatically fix the errors. In the CRL domain, expressions following the grammar are parsed into corresponding formal representations, which contain all essential linguistic information for robotic planning with reference to IEEE CORA standards [15]. As shown in Fig. 1, given a sequence of natural-language instructions, the CRL parser generates the corresponding syntactic structure for expressions without errors in grammar, or corrects fixable patterns for expressions with errors in grammar. The CRL framework will translate the instructions into knowledge representations or robot action planning.

We plan to address two fundamental challenges. One challenge to design a linguistic model that is comprehensive and unambiguous. To the best of our survey, there is no work so far addressing the importance of these two properties equivalently. The other challenge is to automatically transform or represent formal representation into robot knowledge and actions. A primary objective of human-robot communication is to enable high-level mutual understanding and automated planning. The main contributions of this paper are:

1) We designed and implemented a linguistic grammar tailed for robotic applications, which achieves both reliability and expressiveness; and
2) We developed a complete framework and proposed a methodology in translating natural-language instructions into corresponding robot actions.

II. Related work

Assembly in "light" industry involves manipulating small, fragile components and combining them back into a complete and functional object in an unstructured environment. There is a significant interest in assembly problems recently [16]–[19], to construct furniture items using different taxonomy approaches and workspace setup. Typically, a research branch focuses on a bimanual workspace and motion-centric taxonomy showed success in constructing back an IKEA chair step-by-step using two manipulators attached to a fixed and optimized environment [16], [17]. Oppositely, [18] finds interest in using a team of heterogeneous robots, freely moving around the workspace to find their best positions to partially accomplish the predefined plan. Using motion-centric taxonomy generated from geometric reasoning, they demonstrated the building progress for an IKEA table using multiple KUKA youBots. On another hand, [19] instructs robots solving assembly problems by analyzing and learning the patterns from a sequence of specified configurations "taught" by human. Recently, the appearance of a mutual benchmark for furniture assembly problems encourages research in this field, especially in reinforcement learning approaches [20]. However, most work so far has the following common limitations: (i) focus on finding appropriate taxonomy for each furniture object; (ii) primitive actions omit object's characteristics (motion-centric); (iii) a plan is written by computer languages rather than human languages. Our proposed framework will explore problems (ii) and (iii) by using framework sensitive with object-centric taxonomy (where chosen actions are based on object's characteristics) and using natural language interface for planning specification.

NL instruction understanding is the machinery attempts to comprehend human command by mapping correct robotic actions or extracting correct information from the sentence. NL understanding has been studied extensively in the general domain as well as robotic specification. Liu and Zhang provided an excellent review of the latest and state-of-art methodologies in the field [5]. The two most distinguishable and common approaches are *grammar model* and *association model*. In *grammar model*, robots understand commands by analyzing linguistic patterns from predefined grammar. Having an appropriate choice of grammar, NL model can extract deep and complicate linguistic information such as temporal and spatial relations [21], [22], relation between objects and entities' roles [12]. Despite this model could be constructed without the requirement of the training data, *grammar model* has a bounded generality and sensitive to grammatical syntax. *Association model*, on another hand, can handle abstract and implicit NL instructions beyond human language. The key component for such robustness is to use no grammar at all. Rather, grammar is implicitly learned from probabilistic models trained on the pre-collected dataset. *Association model* can recognize main primitive action and locations [23], actions' adjuncts [9], objects' states [8]. Despite its general expressiveness, *association model* requires significant of training data and does not guarantee certainty. Performances of these models will behave differently depending on the located workspace, make them unsuitable for a sensitive environment.

III. The Framework of Controlled Robot Language (CRL)

In view of the lack of a linguistic interface balancing reliability and expressiveness, we aim to implement a framework of controlled robot language (CRL) for robot understanding and planning. Motivated by controlled natural language, the proposed model maintains essential properties: certain and general-purpose expressive. The proposed framework contains three fundamental components: CRL grammar, CRL parser, and CRL translator. The CRL grammar defines a formal language domain, taking a general set of English input and assign syntactic structures. From syntactic structures, essential linguistic information from a sentence such as *subject*, *object*, *predicate*, *noun modifier*, *predicate modifier* can be extracted and constructed back into formal representations. The CRL grammar is designed toward a deterministic and

reliable interpretation with no ambiguity. We limited the set of CRL grammar for a compact and efficient grammar core. The sentences outside the domain are processed by CRL parser, which includes a set of flexible dialect patterns that exist in daily speech but cannot be expressed as a grammar rule. These dialect patterns can be recognized and corrected by the CRL translator, which finds equivalent but valid expressions in CRL.

A. CRL Grammar and Dialect Patterns

Grammar is the core component of *grammar model* approach. We designed the CRL grammar as a general-purpose and deterministic grammar in English. The CRL grammar defines an unambiguous formal language domain, which can be accurately and efficiently processed by a computer, but is still expressive enough to allow natural usage. We defined and developed the CRL grammar in terms of Context Free Grammar (CFG) with selective rules to avoid unnecessary ambiguity.

Given a set of terminal nodes associated with a set of terminal symbols $\mathcal{T}$ and nonterminal nodes associated with set of nonterminal symbols $\mathcal{N}$, we defined grammar using CFG formalism. CFG grammar is a collection of linguistic productions in the form of

$$X \rightarrow \{Y_i\}_i^n \ \{\alpha_j\}_j^m \tag{1}$$

where $X \in \mathcal{N}$, $Y_i \in \mathcal{N} \cup \mathcal{T}$ and $\alpha_j \in \mathcal{T}$. The current version of the CRL is constructed by set of 110 productions[1], which are visualized in Fig. 2. Because of the elegant design, the grammar can capture common linguistic expressions in real-world scenarios, e.g., *"A robot pick a red apple on the table."* and *"Which apple is red?"*. The grammar does have constraints on valid expression, aiming to maximally avoid unnecessary ambiguity.

B. CRL Parser

To improve the expressiveness of the CRL grammar without adding additional rules that can implicitly create ambiguity, we developed a CRL parser containing a set of dialect patterns that commonly appear in communication. These patterns are detected and corrected by the parser through syntax directed translation. The dialect patterns are flexible and dynamic in the linguistic domain, so the CRL parser provides manipulating functions to easily update the patterns to various application domains. We have developed a set of dialect patterns by analyzing WikiHow instructions [24], as shown in Table I.

With the defined CRL grammar, we constructed a syntactic parser to analyze syntactic structures of natural-language descriptions. We utilized a simple dynamic-programming based approach CKY to find a syntactic structure for each sentence. Furthermore, to maintain the reliability of robotic systems, the model does not automatically resolve ambiguity; instead, whenever ambiguity appears, the robotic system asks for user's decisions to disambiguate the sentence. The user's decisions were collected as a database, which were used to train a neural-network preference model that adapts flexibly to different context.

[1]The grammar and parsers are available at: https://github.com/hhelium.

TABLE I
GRAMMAR PATTERNS FOR CRL CHECKER

Dialect Pattern	**Sample**	**Correction**
Imperative	*rotate* the leg	adding *"robot"* as default subject
Compound noun	*table leg*	last noun as main noun
Consecutive adjectives	a *small red* apple	adding conjunction "and"
Consecutive adverbs	*gradually slowly* move	adding conjunction "and"
Your/our/my pattern	*your* hand	replace with valid determiner
You/I/we pattern	*you* can move	replace with valid pronoun
Verb + obj + to + verb	click the screen to start	rephrase the expression
Verb + to + verb)	have to wait	adding possibility-modal
Verb + gerund	consider stopping	gerund as main verb
From-to pattern	*from 0.1 to 0.2 cm*	rephrase the expression
Plural nouns	*cubes*	singularize
Passive voice	is picked by	rephrase the expression
Metrics	3 kg, 3 kilograms, 1 ton	mapping to fixed set of units
Literal quantity	one half; quarter; dozen	mapping to fixed set of units + quantifying
Progress description ("by doing")	*by grasping its hand*	rephrase the expression

IV. AUTOMATED ROBOT PLANNING IN CRL

Leveraging the parsed syntactic structures, the CRL framework constructs a contextual semantic representation of natural-language expressions. These semantic representations are essential for robot understanding and planning. These semantic representations contains three types of information: contextual descriptions, command instructions, and queries. The command instructions corresponds to robot planning, and the contextual descriptions specify work context and constraints; therefore, we investigated the methodology in automatically translating these instructions into corresponding LTL specification, which can be implemented generally by robot systems.

A. Translating CRL Descriptions to Knowledge Representations

Given the syntactic structure, we extracted fundamental linguistic components such as *object*, *predicate*, *property*, *adjunct,* and *phrase*. The extracted information is, however, discrete and exclusive. To unify it into a single semantic representation, we need semantic rules to depict combining procedures. These rules are best described using symbolic language as Prolog [25]. The main challenge of this process is the ability to create useful rules that can unify low-level knowledge. Appropriate selection of semantic rules goes beyond a combination task: solve anaphoric problem, identify quantification property, and describe temporal constraints. In this paper, we focus on a set of semantic rules for discourse

```
Noun_Count → NN
Noun_Count → NNS
Prop → NNP
Prop → NNPS
Mod_Noun → ADJP Noun_Count
Mod_Noun → Noun_Count
NP → Prop
NP → PRP
NP → DT Mod_Noun
NP → Prop POS Mod_Noun
NP → DT Mod_Noun POS Mod_Noun
NP → DT Mod_Noun POS_IN DT Mod_Noun
NP → DT Mod_Noun POS_IN Prop
NP → NP relcl
Adj → JJ
Adj → "more" JJ
Adj → JJR
Adj → "most" JJ
Adj → JJS
ADJP → Adj
Adv → RB
Adv → "more" RB
Adv → RBR
Adv → "most" RB
Adv → RBS
ADVP → Adv
Adj_C → "as" JJ "as" NP
Adj_C → JJR "than" NP
Adj_C → "more" JJ "than" NP
Adj_C → JJC NP
Adj_C → "as" JJC NP "as" NP
Adj_C → "as" JJC NP "as" RP_ADJ NP
Adj_C → JJCR NP
Adj_C → JJCR NP "than" NP
Adj_C → JJCR NP "than" RP_ADJ NP
Adj_C → "more" JJC NP
Adj_C → "more" JJC NP "than" NP
Adj_C → "more" JJC NP "than" RP_ADJ NP
IVerb → VBZ_I
IVerb → VB_I
TVerb → VBZ_T
TVerb → VB_T
TVerb → VBD_T
DVerb → VBZ_D
DVerb → VB_D
DVerb → VBD_D
Main_Verb → IVerb
Main_Verb → TVerb NP
Main_Verb → TVerb "by" NP
Main_Verb → DVerb NP NP
Main_Verb → NP
Main_Verb → ADJP
Main_Verb → Adj_C
Verb_Mod → PP
Verb_Mod → PP Verb_Mod
Verb_Mod → ADVP
Verb_Mod → ADVP Verb_Mod
VP → Main_Verb
VP → Main_Verb Verb_Mod
VP → Verb_Mod Main_Verb
VP → AUX Main_Verb
VP → AUX Main_Verb Verb_Mod
VP → Verb_Mod AUX Main_Verb
VP → AUX Verb_Mod Main_Verb
VP → Verb_Mod Main_Verb Verb_Mod
PP → IN NP
relcl2 → VP
VP_coord_2 → VP "and" VP_coord_2
VP_coord_2 → VP
ADJP → Adj "and" ADJP
ADVP → Adv "and" ADVP
relcl2 → VP and_WhP relcl2
VP_coord_1 → VP_coord_2
VP_coord_1 → VP_coord_2 "or" VP_coord_1
relcl1 → relcl2
relcl1 → relcl2 or_WhP relcl1
relcl → WhP relcl1
simple_sentence_2 → NP VP_coord_1
simple_sentence_1 → simple_sentence_2
simple_sentence_1 → neg_clause simple_sentence_2
simple_sentence_1 → there_clause NP
simple_sentence_1 → there_clause NP such_clause simple_sentence_1
neg_clause →"it" "is" "false" "that"
there_clause → EX "is"
there_clause → EX "are"
such_clause → "such" "that"
sentence_coord_2 → simple_sentence_1
sentence_coord_2 → simple_sentence_1 "and" sentence_coord_2
sentence_coord_1 → sentence_coord_2
sentence_coord_1 → sentence_coord_2 "or" sentence_coord_1
sentence → sentence_coord_1
sentence → for_clause Of_Noun sentence_coord_1
sentence → "if" sentence_coord_1 "then" sentence_coord_1
for_clause → "for" "every"
complete_sentence → sentence "."
complete_sentence → AUX NP Main_Verb "?"
complete_sentence → AUX NP Main_Verb Verb_Mod "?"
complete_sentence → AUX NP Verb_Mod Main_Verb "?"
complete_sentence → AUX NP Verb_Mod Main_Verb Verb_Mod "?"
complete_sentence → WhP VP "?"
complete_sentence → WhDT Mod_Noun VP "?"
complete_sentence → WhDT Mod_Noun AUX NP TVerb "?"
complete_sentence → WhDT Mod_Noun AUX NP TVerb Verb_Mod "?"
complete_sentence → WhDT Mod_Noun AUX NP Verb_Mod TVerb "?"
complete_sentence → WhDT Mod_Noun AUX NP Verb_Mod TVerb Verb_Mod "?"
complete_sentence → WhDT Mod_Noun AUX NP DVerb NP "?"
complete_sentence → WhDT Mod_Noun AUX NP DVerb NP Verb_Mod "?"
complete_sentence → WhDT Mod_Noun AUX NP Verb_Mod DVerb NP "?"
complete_sentence → WhDT Mod_Noun AUX NP Verb_Mod DVerb NP Verb_Mod "?"
complete_sentence → WhADVP AUX NP VP "?"
```

Fig. 2. CFG productions for CRL grammar – implicit descriptions of CRL grammar.

representations, which also handle anaphora binding and determiners quantifying. The theoretical details of these rules can be found at Kamp's lecture [26]. The developed CRL system expanded the semantic rules as defined in [27] by unifying discrete components and adding a few abstraction layers.

These abstraction layers provide convenient ways to identify objects, trigger actions, or query. Fig. 3 shows some examples that represents semantic discourse outputs for CRL valid expressions. By using additional abstraction layers, we can easily classify all possible formal representation into three main types: (i) *Description-type*: used to describe an event, robotic environment, and knowledge base; (ii) *Command-type*: used to trigger robot actions; (iii) *Query-type*: extract information back from the knowledge base.

B. Generate Robot Planning from CRL Instructions

To map *command-type* expressions to atomic robotic actions, we developed a new mapping technique that translates each instruction into an equivalent Logical Temporal Logic (LTL) representations. The LTL representations has demonstrated effectiveness and robustness in robot action planning [4], [28]. This technique can be extended for a sequence of commands describing liveness property in LTL theory. A fragment of LTL language corresponding to robot actions can be represented by $G = \langle \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_e, \theta_s, \rho_e, \rho_s, \varphi \rangle$ [29], where

- $\mathcal{V} = \{v_1, ..., v_n\}$ is a finite set of state variables. Each variable has their own discrete domain.
- $\mathcal{X} \subseteq \mathcal{V}$ is a set of *input variables,* maintained and controlled by environment.
- $\mathcal{Y} \subseteq \mathcal{V} \setminus \mathcal{X}$ is a set of *output variables,* maintained and

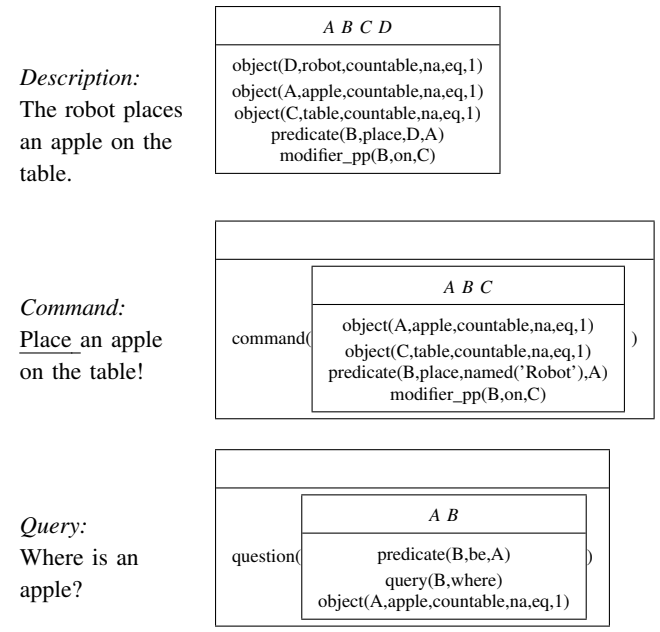

Fig. 3. Example of formal representation of CRL sentences. Each input CRL sentence in the left side corresponding to a formal representation in right side. The formal representation is further classified in either (i) description-type; (ii) command-type; (iii) query type.

controlled by the system.

- θ_e is an assertion over $\mathcal{X}$ characterizing initial states of the environment.
- θ_s is an assertion over $\mathcal{V}$ characterizing initial state of the system.
- $\rho_e(\mathcal{V}, \mathcal{X}')$ is transition description of the environment. This is assertion relating a current state variable $s \in \mathcal{V}$ and its primed version $s' \in \mathcal{X}'$, where primed variable indicate itself in the "next" cycle.
- $\rho_s(\mathcal{V}, \mathcal{X}', \mathcal{Y}')$ is transition description of the system. This is assertion relating a current state variable $s \in \mathcal{V}$, its system "next" state $s' \in \mathcal{Y}'$ and its environment "next" state $s'_e \in \mathcal{X}'$.

We defined each atomic action as a disjunction of transitions, where arguments of an action are described as environment variables. Following this procedure, we can theoretically describe arbitrary action concepts in term of LTL specification. Given a set of action transitions $\rho_s(\mathcal{V}, \mathcal{X}', \mathcal{Y}')$, where each transition is a conjunction of current state and next state properties, a transition from the instruction state s_A to state s_B can be written as

$$(s = A) \wedge \bigcirc(s = B) \tag{2}$$

where $s \in \mathcal{V}$, $A, B \in domain(s)$, and $\bigcirc$ is a temporal logic operator indicates next cycle.

Transition is a most primitive movement that robot system can operate; however, it is not an action used in linguistic command. From a logic perspective, a concept action is an abstraction and quantified form of transition, which we describe as disjunction of transitions. More specifically, the general formalism for an arbitrary action without argument is

$$\langle \text{action} \rangle : \bigvee_i \big((s = s_{cur}) \wedge \bigcirc(s = s_{next})\big) \tag{3}$$

where $s \in \mathcal{V}$, $s_{cur}, s_{next} \in domain(s)$, and $\bigcirc$ is a temporal logic operator indicates next cycle. For actions containing arguments, we consider them as *preconditions* of current state that needed to be checked before executing the action. The formalism is in the form

$$\langle \text{action}(x) \rangle : \bigvee_i \big((x = x_{pre}) \wedge (s = s_{cur}) \wedge \bigcirc(s = s_{next})\big) \tag{4}$$

where $x \in \mathcal{V}, x_{pre} \in domain(x)$ is a single argument of the action. The procedure can extend for multiple arguments case without lost of generality. For examples, to assemble four legs into four designated positions, we define the atomic action "place", which is an movement starting with arm's position at default pose, and ending with one of four possible predefined locations

$$\begin{aligned} \text{place}: \quad & \vee(position = default) \wedge \bigcirc(position = area1) \\ & \vee(position = default) \wedge \bigcirc(position = area2) \\ & \vee(position = default) \wedge \bigcirc(position = area3) \\ & \vee(position = default) \wedge \bigcirc(position = area4) \end{aligned}$$

Each of the primitive actions contain an equivalent LTL formulation, which can be implemented on all robot systems supporting the capabilities.

V. Experiment

In the experiments, we evaluated the proposed CRL framework and compared it with other linguistic frameworks on natural-language instruction corpora [30]. We also demonstrated the effectiveness of the CRL framework in automated robot planning in assembling an IKEA furniture table. The experiment was conducted by using a Sawyer manipulator with a integrated anthropomorphic hand [31].

A. Parsing Natural Language by the CRL Framework

The CRL Framework was evaluated on two practical human-friendly corpus: WikiHow [24] and Collaborative Manipulator corpus [32]. The WikiHow dataset has been commonly used in robotic research [12]. The WikiHow dataset is a collection of human describing procedural task using step-by-steps instruction style. Thought it was designed for general applications, the WikiHow dataset is an excellent corpus that provides a wide variety of intuitive instructions that human can understand and accomplish easily.

Among $180,000$ *HowTo* articles, we chose 1386 articles covering general topics such as cooking instructions. The data consists of 9520 instruction sentences and includes 75733 tokens distributing over 3421 words vocabulary. We implemented the same experiment on Collaborative Manipulation corpus, which is a smaller size robotic corpus focusing on manipulation tasks. The CRL framework was compared with the existed ACE framework [30], whose grammar is designed for general semantic representation.

As the results in Table II show, the CRL is outperforming in expressiveness, especially in intuitive linguistic commands. Though defined in general language domain, the ACE framework cannot understand any sentences in WikiHow corpus, due to the extensively use of invalid expressions. Meanwhile, the CRL framework has a flexible set of linguistic patterns that can be detected and corrected, allowing flexible adaption to different scenarios. The performance shows the same trail in the Collaborative Manipulation corpus. More importantly, both CRL and ACE frameworks guarantees deterministic and reliability. The system behavior is always certainty and predictable, which is critical for robotic system.

TABLE II
PERFORMANCE COMPARISON.

		WikiHow (9520)	Collaborative Manipulation (1670)
CNL	**Parsable**	0 (0%)	140 (8.14%)
	Unparsable	9520 (100%)	1534 (91.85%)
	Deterministic	100%	100%
CRL	**Parsable**	3898 (40.94%)	739 (44.25%)
	Unparsable	5621 (59.04%)	931 (55.74%)
	Deterministic	100%	100%

B. Robot Planning Under the CRL Framework

We implemented an automated system based on CRL for the assembly of an IKEA table by following natural-language instructions. The robotic system consists of a Sawyer manipulator with an AR10 robotic hand, which support context-aware task-oriented manipulation [31]. We developed a simulated system in ROS. We employed the RRTConnect [33] planner for low-level control, and developed a Python modules for the CRL interface. The control and communication are implemented as ROS services. We also utilized the Spot [34] to handle LTL specification by building reactive system from DRS instructions, and used RViz for simulation and visualization.

An example of successful execution of action sequences is shown in Fig. 4. The experiment showed that CRL can successfully understand natural-language planning instructions to assemble an IKEA table step by step. Fig. 4 demonstrates four four primitive actions: pick, place, release, and rotate. The choice of action depends on the resulting DRS command instructions. The complete implementation of 13 instruction scripts takes about 4 minutes to finish, but there is a lot of rooms for optimization both in CRL parsing and action planning.

VI. CONCLUSION

In this paper, we proposed a CRL framework that ensures reliability and expressiveness for natural language interface. We also demonstrated the procedure to integrate the CRL framework into robotic planning: from building a complete semantic representation to mapping those representation into robotic actions. The experiment showed the outperformance of the CRL frame in parsing natural-language instructions, and demonstrated the effectiveness and flexibility of the CRL framework for automated robot planning.

REFERENCES

[1] R. E. Yagoda and D. J. Gillan, "You want me to trust a robot? the development of a human–robot interaction trust scale," *International Journal of Social Robotics*, vol. 4, pp. 235–248, Aug 2012.

[2] D. Jain, L. Mosenlechner, and M. Beetz, "Equipping robot control programs with first-order probabilistic reasoning capabilities," in *2009 IEEE International Conference on Robotics and Automation*, pp. 3626–3631, 2009.

[3] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn, "What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution," pp. 4163 – 4168, 06 2009.

[4] C. Finucane, Gangyuan Jing, and H. Kress-Gazit, "Ltlmop: Experimenting with language, temporal logic and robot control," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1988–1993, 2010.

[5] R. Liu and X. Zhang, "Methodologies for realizing natural-language-facilitated human-robot cooperation: A review," *CoRR*, vol. abs/1701.08756, 2017.

[6] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Twenty-fifth AAAI conference on artificial intelligence*, 2011.

[7] G. Salvi, L. Montesano, A. Bernardino, and J. Santos-Victor, "Language bootstrapping: Learning word meanings from perception-action association," *CoRR*, vol. abs/1711.09714, 2017.

[8] Jianwei Zhang and A. Knoll, "A two-arm situated artificial communicator for human-robot cooperative assembly," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 651–658, 2003.

[9] M. Brenner, N. Hawes, J. Kelleher, and J. Wyatt, "Mediating between qualitative and quantitative representations for task-orientated human-robot interaction.," pp. 2072–2077, 01 2007.

[10] Y. Bisk, D. Yuret, and D. Marcu, "Natural language communication with robots," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 751–761, 2016.

[11] S. Lauria, G. Bugmann, T. Kyriacou, J. Bos, and A. Klein, "Training personal robots using natural language instruction," *IEEE Intelligent systems*, vol. 16, no. 5, pp. 38–45, 2001.

[12] M. Tenorth and M. Beetz, "Knowrob: A knowledge processing infrastructure for cognition-enabled robots," *International Journal of Robotics Research*, vol. 32, pp. 566–590, 04 2013.

[13] H. Kamp, J. Van Genabith, and U. Reyle, "Discourse representation theory," in *Handbook of philosophical logic*, pp. 125–394, Springer, 2011.

[14] N. H. Kirk, D. Nyga, and M. Beetz, "Controlled natural languages for language generation in artificial cognition," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6667–6672, 2014.

[15] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, *et al.*, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1193–1204, 2013.

[16] F. Suárez-Ruiz and Q. Pham, "A framework for fine robotic assembly," *CoRR*, vol. abs/1509.04806, 2015.

[17] P. Lertkultanon and Q.-C. Pham, "A certified-complete bimanual manipulation planner," 2017.

[18] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "Ikeabot: An autonomous multi-robot coordinated furniture assembly system," in *2013 IEEE International Conference on Robotics and Automation*, pp. 855–862, 2013.

[19] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Robot Programming by Demonstration*, pp. 1371–1394. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[20] Y. Lee, E. S. Hu, Z. Yang, A. Yin, and J. J. Lim, "Ikea furniture assembly environment for long-horizon complex manipulation tasks," 2019.

[21] H. Zender, P. Jensfelt, O. Mozos, G.-J. Kruijff, and W. Burgard, "An integrated robotic system for spatial understanding and situated interaction in indoor environments.," vol. 2, pp. 1584–1589, 01 2007.

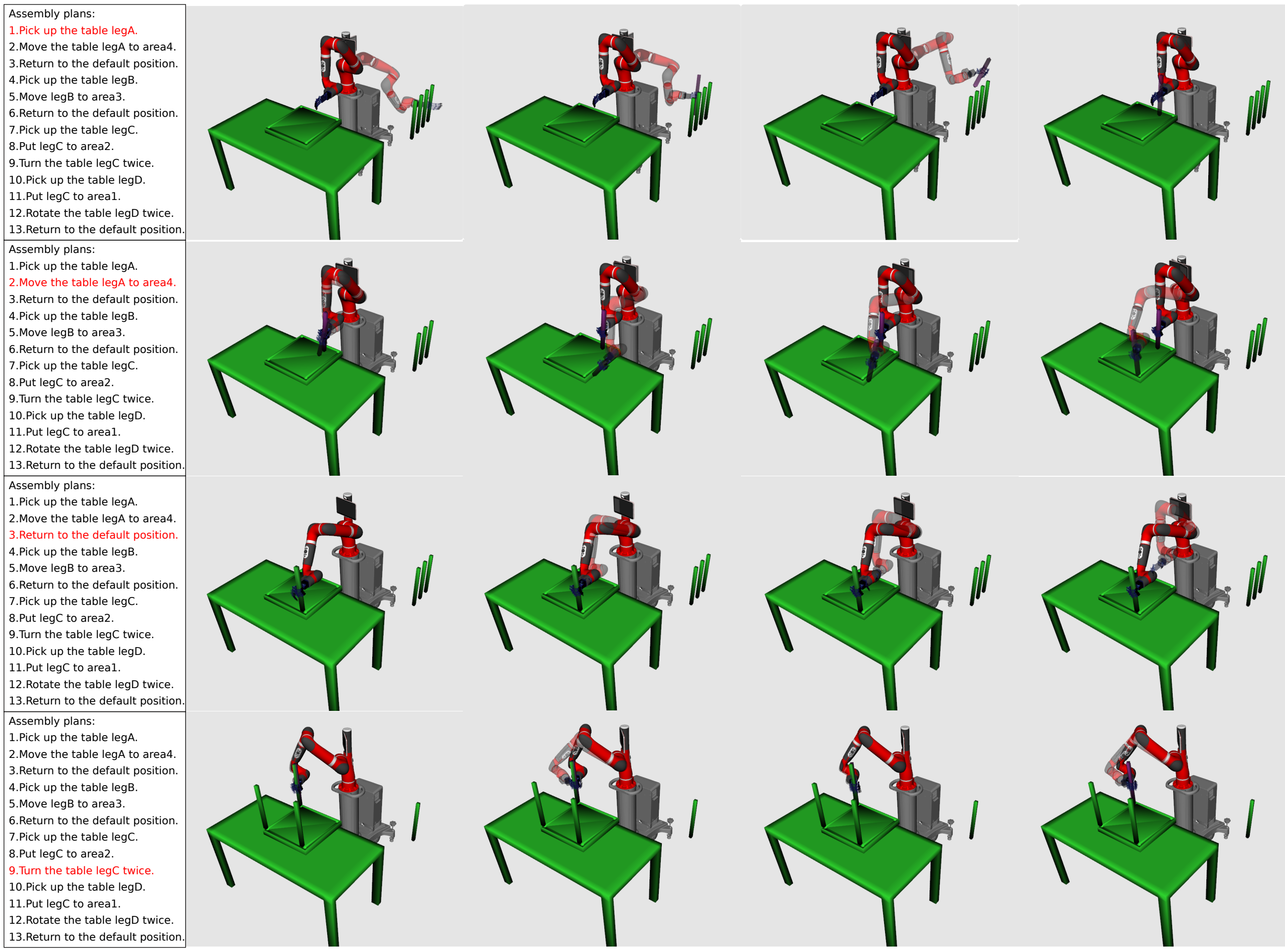

Fig. 4. Applying CRL framework in representing different forms of semantic information. At the upper-left corner, command is input by the user. CRL analyzes instruction and returns semantics for each perspective of robot's perception in DRS form. Different information pieces are included such as *predicate, object, property.*

[22] S. Guadarrama, L. Riano, D. Golland, D. GoÂšhring, Y. Jia, D. Klein, P. Abbeel, and T. Darrell, "Grounding spatial relations for human-robot interaction," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1640–1647, 2013.

[23] T. Kollar, V. Perera, D. Nardi, and M. Veloso, "Learning environmental knowledge from task-based human-robot dialog," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4304–4309, 2013.

[24] M. Koupaee and W. Y. Wang, "Wikihow: A large scale text summarization dataset," 2018.

[25] I. Bratko, *Prolog programming for artificial intelligence*. Pearson education, 2001.

[26] H. Kamp, J. Genabith, and U. Reyle, *Discourse Representation Theory*, pp. 125–394. 11 2010.

[27] T. Kuhn, *Controlled English for knowledge representation*. PhD thesis, University of Zurich, 2010.

[28] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[29] N. Piterman, A. Pnueli, and Y. Saar, "Synthesis of reactive (1) designs," in *International Workshop on Verification, Model Checking, and Abstract Interpretation*, pp. 364–380, Springer, 2006.

[30] N. E. Fuchs and R. Schwitter, "Attempto controlled english (ace)," *arXiv preprint cmp-lg/9603003*, 1996.

[31] H. Li, J. Tan, and H. He, "Magichand: Context-aware dexterous grasping using an anthropomorphic robotic hand," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9895–9901, 2020.

[32] R. Scalise, S. Li, H. Admoni, S. Rosenthal, and S. S. Srinivasa, "Natural language instructions for human–robot collaborative manipulation," *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 558–565, 2018.

[33] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 995–1001, IEEE, 2000.

[34] A. Duret-Lutz, A. Lewkowicz, A. Fauchille, T. Michaud, E. Renault, and L. Xu, "Spot 2.0 — a framework for LTL and ω-automata manipulation," in *Proceedings of the 14th International Symposium on Automated Technology for Verification and Analysis (ATVA'16)*, vol. 9938 of *Lecture Notes in Computer Science*, pp. 122–129, Springer, Oct. 2016.