

Grounding Complex Natural Language Commands for Temporal Tasks in Unseen Environments: Supplementary Material

Anonymous Author(s)

Affiliation

Address

email

1 Specification Patterns

We developed Lang2LTL to ground navigational commands to LTL formulas. We started with the catalog of robotic mission-relevant LTL patterns for robotic missions by Menghi et al. [1]. We adopted 15 templates that are relevant to robot navigation, and modified some of their patterns to semantically match our requirements. The complete list of pattern descriptions and the corresponding LTL templates is in Table 4. Note that we use some additional abbreviated temporal operators, specifically “Weak until” \mathbf{W} , and the “Strong release” \mathbf{M} . In terms of standard operators $a \mathbf{W} b = a \mathbf{U} (b \vee \mathbf{G}a)$, and $a \mathbf{M} b = b \mathbf{U} (a \wedge b)$.

2 Semantic Information from OpenStreetMap Database

We show an examples of semantic dataset as follows,

```
{
  "Jiaho supermarket": {
    "addr:housenumber": "692",
    "shop": "supermarket",
    "opening_hours": "Mo-Su 08:00-20:00",
    "phone": "6173389788",
    "addr:postcode": "02111",
    "addr:street": "Washington Street"
  },
  ...
}
```

3 Implementation Details about Referring Expression generation

We prompt GPT-4 model for paraphrasing landmarks with corresponding OSM databases. For each landmark, three referring expressions are generated and stored for future usage. The prompt for

25 generating referring expressions by paraphrasing is as follows,

Use natural language to describe the landmark provided in a python dictionary form in a short phrase.

Landmark dictionary:

'Fortuna Cafe': {'addr:housenumber': '711', 'cuisine': 'chinese', 'amenity': 'restaurant', 'addr:city': 'Seattle', 'addr:postcode': '98104', 'source': 'King County GIS;data.seattle.gov', 'addr:street': 'South King Street'}

Natural language:

Chinese cafe on South King Street

Landmark dictionary:

'Seoul Tofu House & Korean BBQ': {'addr:housenumber': '516', 'cuisine': 'korean', 'amenity': 'restaurant', 'addr:city': 'Seattle', 'addr:postcode': '98104', 'source': 'King County GIS;data.seattle.gov', 'addr:street': '6th Avenue South'}

Natural language:

Seoul Tofu House

Landmark dictionary:

'AI Video': {'shop': 'electronics'}

Natural language:

AI Video selling electronics

26

...

Landmark dictionary:

'Dochi': {'addr:housenumber': '604', 'cuisine': 'donut', 'amenity': 'cafe'}

Natural language:

a cafe selling donut named Dochi

Landmark dictionary:

'AVA Theater District': {'addr:housenumber': '45', 'building': 'residential', 'building:levels': '30'}

Natural language:

AVA residential building

Landmark dictionary:

'HI Boston': {'operator': 'Hosteling International', 'smoking': 'no', 'wheelchair': 'yes', 'tourism': 'hostel'}

Natural language:

HI Boston

Landmark dictionary:

27 4 Implementation Details about Referring Expression Recognition

28 The prompt for referring expression recognition is as follows,

Your task is to repeat exact strings from the given utterance which possibly refer to certain propositions.

Utterance: move to red room
Propositions: red room

Utterance: visit Cutler Majestic Theater
Propositions: Cutler Majestic Theater

Utterance: robot c move to big red room and then move to green area
Propositions: big red room | green area

Utterance: you have to visit Panera Bread on Beacon Street, four or more than four times
Propositions: Panera Bread on Beacon Street

29 Utterance: go to Cutler Majestic Theater at Emerson College on Tremont Street, exactly three times
Propositions: Cutler Majestic Theater at Emerson College on Tremont Street

...

Utterance: make sure you never visit St. James Church, a Christian place of worship on Harrison Avenue, Dunkin' Donuts, Thai restaurant Montien, New Saigon Sandwich, or Stuart St @ Tremont St
Propositions: St. James Church, a Christian place of worship on Harrison Avenue | Dunkin' Donuts | Thai restaurant Montien | New Saigon Sandwich | Stuart St @ Tremont St

Utterance: move the robot through yellow region or small red room and then to large green room
Propositions: yellow region | small red room | large green room

Utterance:

30 5 Implementation Details about Lifted Translation

31 5.1 Fintuned T5-Base

32 For finetuning T5-base model, we set batch size equal to 40, learning rate equal to 10^{-4} , and weight
33 decay equal to 10^{-2} . We ran training for 10 epochs and picked the best-performing one for reporting
34 results.

35 5.2 Finetuned GPT-3:

36 The per specification type accuracies and the accuracies for varying number of propositions in the
37 formula while testing the finetuned GPT-3 model on the utterance holdout is depicted in Figure ??.
38 The finetuned GPT-3 model achieves high accuracies across formula types and varying numbers of
39 propositions. It shows the benefit of having a large high-quality dataset of natural language commands
40 representing diverse LTL formulas. All previous works also used utterance holdout as their testing
41 methodology, but their training and test sets contains significantly less unique LTL formulas.

42 5.3 Prompt GPT-4

43 The prompt for end-to-end GPT-4 is as follows,

Your task is to translate English utterances into linear temporal logic (LTL) formulas.

Utterance: visit b

LTL: $F b$

Utterance: eventually reach b and h

LTL: $\& F b F h$

Utterance: go to h a and b

44 LTL: $\& F h \& F a F b$

Utterance: proceed to reach h at the next time instant only and only if you see b

LTL: $G e b X h$

Utterance: wait at b till you see h

LTL: $U b h$

Utterance: go to h in the very next time instant whenever you see b

LTL: $G i b X h$

Utterance:

45 5.4 Prompt GPT-3

46 The prompt for end-to-end GPT-3 is the same as the one we used for Prompt GPT-4.

47

48 5.5 Seq2Seq Transformer

49 We constructed and trained a transformer model following [2]. More specifically, we built the model's
50 encoder with three attention layers and decoder with three layers, and we used 512 as embedding size
51 and 8 as the number of attention heads. For training, we adapted batched training with a batch size
52 equal to 128, learning rate equal to 10^{-4} , and dropout ratio equal to 0.1; the training process runs for
53 10 epochs, and we picked the best-performing checkpoint for baseline comparison.

54 5.6 Type constrained decoding (TCD)

55 constrained decoding has been used in generating formal specifications for eliminating syntactically
56 invalid outputs. Due to the sampling nature of NN-based models, generated tokens from the output
57 layer can result in syntactical errors that can be detected on the fly, and type-constrained decoding
58 solves it by forcing the model to only generate tokens following the correct grammar rule. By
59 eliminating syntax errors, it also improves the overall performance of the system.
60 In practice, type-constrained decoding is implemented at each step of the decoding loop: first
61 checking the validity of the output token, then appending the valid token or masking the invalid and
62 re-generating a new token according to the probability distribution after masking. In addition, we
63 design an algorithm to simultaneously enforce the length limitation and syntactical rule by parsing
64 partial formulas into binary trees. Beyond a given maximum height of the tree, the model is forced
65 only to generate propositions but not operators.

66 6 Implementation Details about Code as Policies

67 We designed two prompts for reproducing Code as Policies: one for code generation and the other for
68 parsing landmarks. The code generation prompt is expected to generate an executable python script
69 that calls the goto_loc() function for traversing through the environment and psrse_loc() function to
70 ground referring expressions to landmarks, where the landmark resolution prompt is used. The code
71 generation prompt for graph search is as follows,

```
72 # Python 2D robot navigation script

import random
from utils import goto_loc, parse_loc

# make the robot go to wooden desk.
target_loc = parse_loc('wooden desk')
goto_loc(target_loc)
# go to brown desk and then white desk.
target_loc_1 = parse_loc('brown desk')
target_loc_2 = parse_loc('white desk')
target_locs = [target_loc_1, target_loc_2]
for target_loc in target_locs:
    goto_loc(target_loc)
# head to doorway, but visit white kitchen counter before that.
target_loc_1 = parse_loc('white kitchen counter')
target_loc_2 = parse_loc('doorway')
target_locs = [target_loc_1, target_loc_2]
for target_loc in target_locs:
    goto_loc(target_loc)
# avoid white table while going to grey door.
target_loc = parse_loc('grey door')
73 avoid_loc = parse_loc('white table')
target_locs = [target_loc]
avoid_locs = [avoid_loc]
for loc in target_locs:
    goto_loc(loc, avoid_locs=avoid_locs)
# either go to steel gate or doorway
target_loc_1 = parse_loc('steel gate')
target_loc_2 = parse_loc('doorway')
target_locs = [target_loc_1, target_loc_2]
target_loc = random.choice(target_locs)
goto_loc(target_loc)

...

# go to doorway three times
target_loc = parse_loc('doorway')
for _ in range(3):
    goto_loc(target_loc)
    random_loc = target_loc
    while random_loc == target_loc:
        random_loc = random.choice(locations)
    goto_loc(random_loc)
```

74 The landmark resolution prompt is as follows,

```
# Python parsing phrases to locations script

locations = ['bookshelf', 'desk A', 'table', 'desk B', 'doorway', 'kitchen counter', 'couch',
'door']
semantic_info = {
    "bookshelf": {"material": "wood", "color": "brown"},
    "desk A": {"material": "wood", "color": "brown"},
    "desk B": {"material": "metal", "color": "white"},
    "doorway": {},
    "kitchen counter": {"color": "white"},
    "couch": {"color": "blue", "brand": "IKEA"},
    "door": {"material": "steel", "color": "grey"},
    "table": {"color": "white"},
}
# wooden brown bookshelf
ret_val = 'bookshelf'

...

locations = ['bookshelf', 'desk A', 'table', 'desk B', 'doorway', 'kitchen counter', 'couch',
'door']
semantic_info = {
    "bookshelf": {"material": "wood", "color": "brown"},
    "desk A": {"material": "wood", "color": "brown"},
    "desk B": {"material": "metal", "color": "white"},
    "doorway": {},
    "kitchen counter": {"color": "white"},
    "couch": {"color": "blue", "brand": "IKEA"},
    "door": {"material": "steel", "color": "grey"},
    "table": {"color": "white"},
}
# blue IKEA couch
ret_val = 'couch'
```

76 7 Implementation Details about Grounded Translation

77 7.1 CopyNet

78 For reproducing [3], we trained the CopyNet baseline with our grounded dataset preprocessed as its
79 required format. To make a fair comparison on generalization ability, the CopyNet model has only
80 seen utterance-formula pairs from the Boston subset, and the evaluation is run on grounded datasets
81 of the rest 21 cities. For training CopyNet, we followed closely to the instruction of the original paper
82 and used the exact same LSTM model structure and pre-computed glove embedding for landmark
83 resolution. On the hyperparameters, we set embedding size equal to 128, hidden size equal to 256,
84 learning rate equal to 10^{-3} , batch size equal to 100.

85 7.2 Prompt GPT-4

86 The prompt for end-to-end GPT-4 is as follows. While we tried including a landmark list in the
87 prompt, it was removed in the final version, because we observed empirically that Prompt GPT-

Table 1: Dataset Comparison

	Lang2LTL Lifted	CleanUp World	NL2TL	Wang et al. [4]
Number of datapoints	49655	3382	39367	6556
Unique formula skeletons	47	4	605	45
#Propositions (min, max, mean)	(1,5,3.79)	(1,3,1.85)	(1,7,2.86)	(1,4,2.01)
Formula Length (min, max, mean)	(2,67,18.89)	(2,7,4.77)	(1,13,5.98)	(3,7,4.48)

88 4 achieved better performance without explicitly giving a list of landmarks during prompt engineering.

Your task is to first find referred landmarks from a given list then use them as propositions to translate English utterances to linear temporal logic (LTL) formulas.

Utterance: visit Panera Bread sandwich fast food on Stuart Street
LTL: F panera_bread

Utterance: eventually reach Wang Theater, and The Kensington apartments
LTL: & F wang_theater F the_kensington

89

...

Utterance: make sure that you have exactly three separate visits to Seybolt Park
LTL: M & seybolt_park F & ! seybolt_park F & seybolt_park F & ! seybolt_park F seybolt_park | ! seybolt_park G | seybolt_park G | ! seybolt_park G | seybolt_park G | ! seybolt_park G | seybolt_park G ! seybolt_park

Utterance:

90 8 Dataset Details

91 8.1 Quantifying diversity of temporal commands

92 We quantify the diversity of the temporal commands a system is tested on using the temporal
93 formula skeletons in evaluation corpus of commands. We propose that each novel dataset should be
94 characterized along the following dimensions, and as an example we provide the respective values
95 for the Lang2LTL dataset (lifted and grounded (OSM)) described in Section 6.4 of the main paper.

- 96 1. Number of semantically unique formulas: 47
- 97 2. Number of propositions per formula: minimum: 1, maximum: 5, average: 3.79
- 98 3. Length of formulas: minimum: 2, maximum: 67. average: 18.89
- 99 4. Vocabulary size (for grounded datasets):
- 100 5. Linguistic diversity of utterances: self-BLEU score: 0.85

101 Table 1 compares our proposed lifted dataset and other datasets proposed in prior work.

102 9 Detailed Result Analysis on Lifted Translation

103 We further analyzed the results for each model and holdout type for the lifted translation problem. In
104 particular, we computed the accuracies per each formula type, and the number of unique propositions
105 required to construct the target formula. This analysis provides insights on the sensitivity of the
106 models to particular templates and formula lengths.

107 The accuracies of each model and holdout type categorized by formula types are depicted in Figure 1.
108 We observe that for both the finetuned models (Finetuned T5 and Finetuned GPT-3), the model achieves
109 high accuracies across various formula types for *Utterance Holdout*. Note that the performance

110 across types is more uniform for Finetuned GPT-3 than Finetuned T5-Base model. Next we note that
 111 Prompt GPT-4 achieves better accuracies as compared to Prompt GPT-3 across all evaluations.

112 We observe that the performance of the finetuned models is more unbalanced across different formula
 113 types for the *Formula Holdout* test case. In comparison, Prompt GPT models, achieve non-zero
 114 accuracies across all formula types. Once again Prompt GPT-4 outperforms Prompt GPT-3. We
 115 note that adding type constrained decoding to Finetuned T5-Base during inference only marginally
 116 improved *Utterance Holdout*, but significantly improved *Formula* and *Type Holdout*, which implies
 117 Finetuned T5-Base model is more likely to produce syntactically incorrect output when the grounding
 118 formula instance or type have not seen during training.

119 Finally, we note that only the prompt GPT models achieve meaningful accuracies in the *Type Holdout*
 120 scenarios. However, even in *Type Holdout*, the accuracies are concentrated on formula types that
 121 only had short lengths, or shared subformulas with types seen during training. We can conclude that
 122 *Formula* and *Type Holdout* remain challenging paradigms of generation and an open problem for
 123 automated translation of language commands into formal specifications.

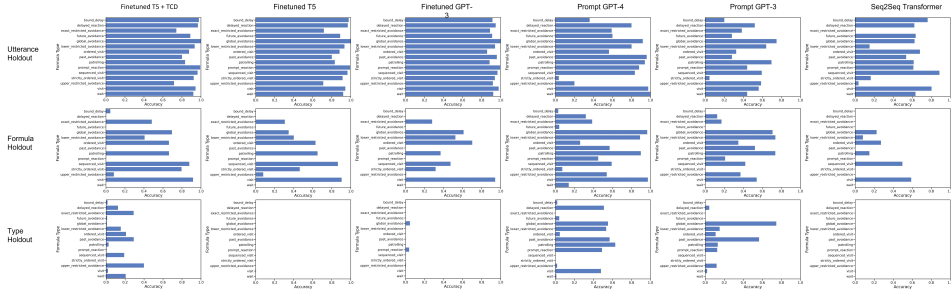


Figure 1: The accuracies per grounding formula types of six lifted translation models

124 Next, we repeated the above analysis, but categorized accuracies by the number of unique propositions
 125 that appear within a formula. The results are depicted in Figure 2.

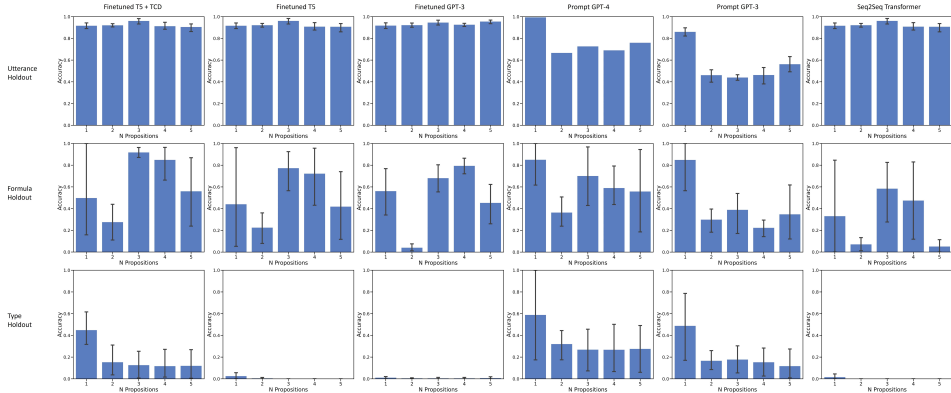


Figure 2: The accuracies per number of unique propositions of six lifted translation models

126 Here we note that in the *Utterance Holdout* test, the finetuned models demonstrated balanced perfor-
 127 mance across the dataset, whereas both prompt GPT models demonstrated degraded performance
 128 when the number of unique propositions in the formula were increasing. Subsequently, the degraded
 129 performance on longer formulas was apparent even within the *Formula* and *Type holdout* domains.
 130 In contrast the three finetuned models performed better for longer formulas in *Formula Holdout*.
 131 We hypothesize that this is because the finetuned models were more able to generalize to different
 132 formula lengths of the same template (in particular the templates that required temporal ordering
 133 constraints to be encoded) as compared to the prompt completion-based approaches. In addition,
 134 there are more samples in the training set for longer formulas due to permutations of propositions.

As finetuning an LLM on the target task produced the best results for *Utterance Holdout*, we further analyzed the cause of errors for the instances where the lifted translation was incorrect. We categorize the errors as follows:

1. **Syntax Errors:** The formula returned by the lifted translation module was not a valid LTL formula.
2. **Misclassified formula type:** The lifted translation module returns an identifiable but incorrect formula type that did not correspond to the input command.
3. **Incorrect propositions:** The returned formula was of the correct formula type but had the incorrect number of propositions.
4. **Incorrect permutation:** The formula was of the correct template class, and had the right number of propositions, but the propositions were in the wrong location within the formula.
5. **Unknown template** The returned formula was a valid LTL formula, but did not belong to any known formula types.

Figure 3 to Figure 5 depict the relative frequencies of the error cases as a pie chart for the three finetuned models. Note that returning unknown formula templates with the correct syntax was the most common cause of error in the lifted translation based on all finetuned models.

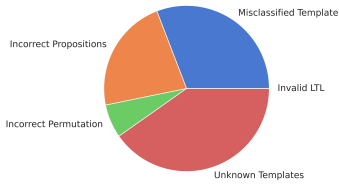


Figure 3: Error frequencies of Finetuned T5-Base with TCD

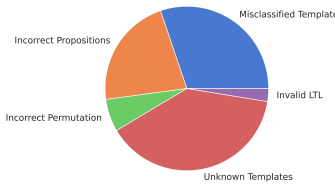


Figure 4: Error frequencies of Finetuned T5-Base

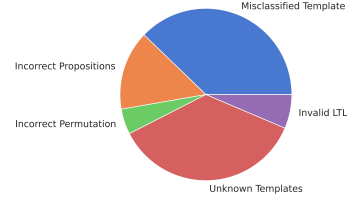


Figure 5: Error frequencies of Finetuned GPT-3

Since Finetuned GPT-3 achieves the best generalization across formula types, and type constrained decoding (TCD) during inference significantly improve the translation accuracies for unseen formula instances and types, the combination of large language models and TCD is by far the best approach for grounding language commands for temporal tasks.

10 Robot Demonstration

10.1 Indoor Environment #1

The semantic information of landmarks in the first household environment is as follows,

```
{
  "bookshelf": {
    "material": "wood",
    "color": "brown"
  },
  "desk A": {
    "material": "wood",
    "color": "brown"
  },
  "desk B": {
    "material": "metal",
    "color": "white"
  },
  "doorway": {},
  "kitchen counter": {
    "color": "white"
  }
}
```

```

174     },
175     "couch": {
176         "color": "blue",
177         "brand": "IKEA"
178     },
179     "door": {
180         "material": "steel",
181         "color": "grey"
182     },
183     "table": {
184         "color": "white"
185     }
186 }

```

187 Natural language commands used to test our proposed system and Code as Policies [5] are shown in
188 Table 2.

189 10.2 Indoor Environment #2

190 The semantic information of landmarks in the second household environment is as follows,

```

191 {
192     "hallway A": {
193         "decoration": "painting"
194     },
195     "hallway B": {
196         "decoration": "none"
197     },
198     "table A": {
199         "location": "kitchen",
200         "material": "metal",
201         "color": "blue"
202     },
203     "table B": {
204         "location": "atrium",
205         "material": "metal",
206         "color": "white"
207     },
208     "classroom": {
209         "door": ["glass", "grey"]
210     },
211     "elevator": {
212         "color": "purple"
213     },
214     "staircase": {},
215     "front desk": {},
216     "office": {
217         "door": ["wood", "yellow"]
218     },
219 }

```

220 Natural language commands used to test our proposed system and Code as Policies [5] are shown in
221 Table 2.

References

- [1] C. Menghi, C. Tsigkanos, P. Pelliccione, C. Ghezzi, and T. Berger. Specification patterns for robotic missions. *IEEE Transactions on Software Engineering*, 47(10):2208–2224, oct 2021. ISSN 1939-3520. doi:10.1109/TSE.2019.2945329.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [3] M. Berg, D. Bayazit, R. Mathew, A. Rotter-Aboyoun, E. Pavlick, and S. Tellex. Grounding Language to Landmarks in Arbitrary Outdoor Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [4] C. Wang, C. Ross, Y.-L. Kuo, B. Katz, and A. Barbu. Learning a natural-language to ltl executable semantic parser for grounded robotics. In *Conference on Robot Learning*, pages 1706–1718. PMLR, 2021.
- [5] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *arXiv preprint arXiv:2209.07753*, 2022.

Table 2: Commands for Robot Demonstration in Indoor Environment #1

Navigational Command	Lang2LTL Result	Code as Policies Result
1. go to brown bookshelf, metal desk, wooden desk, kitchen counter, and the blue couch in any order	success	success
2. move to grey door, then bookrack, then brown desk, then counter, then white desk	success	success
3. visit brown wooden desk but only after bookshelf	success	misunderstand the task
4. go from brown bookshelf to white metal desk and only visit each landmark one time	success	misunderstand the task
5. go to brown wooden desk exactly once and do not visit brown desk before bookshelf	success	inexecutable
6. go to white desk at least three times	success	inexecutable
7. go to wooden bookshelf at least five times	success	success
8. visit bookshelf at most three times	success	success
9. visit counter at most 5 times	success	success
10. go to wooden desk exactly three times	success	misunderstand the task
11. move to brown wooden desk exactly 5 times	success	inexecutable
12. go to doorway exactly two times, in addition always avoid the table	success	success
13. go to brown desk only after visiting bookshelf, in addition go to brown desk only after visiting white desk	success	misunderstand the task
14. visit wooden desk exactly two times, in addition do not go to wooden desk before bookrack	success	inexecutable
15. visit wooden desk at least two times, in addition do not go to wooden desk before bookshelf	success	inexecutable
16. visit the blue IKEA couch, in addition never go to the big steel door	success	success
17. visit white kitchen counter then go to brown desk, in addition never visit white table	success	success
18. go to the grey door, and only then go to the bookshelf, in addition always avoid the table	success	misunderstand the task
19. go to kitchen counter then wooden desk, in addition after going to counter, you must avoid white table	success	misunderstand the task
20. Go to bookshelf, alternatively go to metal desk	success	misunderstand the task
21. Go to counter, alternatively go to metal desk	success	misunderstand the task
22. Go to the counter, but never visit the counter	unsatisfiable. abort correctly	stop execution correctly
23. do not go to the wooden desk until bookshelf, and do not go to bookshelf until wooden desk	unsatisfiable. abort correctly	stop execution correctly
24. go to brown desk exactly once, in addition go to brown desk at least twice	unsatisfiable. abort correctly	misunderstand the task
25. find the kitchen counter, in addition avoid the doorway	unsatisfiable. abort correctly	stop execution correctly
26. move to couch exactly twice, in addition pass by counter at most once	unsatisfiable. abort correctly	stop execution correctly
27. navigate to the counter then the brown desk, in addition after going to the counter, you must avoid doorway	unsatisfiable. abort correctly	misunderstand the task
28. Visit the counter at least 2 times and at most 5 times	incorrect grounding. OOD	inexecutable
29. visit counter at least six times	incorrect grounding. OOD	success
30. either go to bookshelf then desk A, or go to couch	incorrect grounding. OOD	misunderstand the task

Table 3: Commands for Robot Demonstration in Indoor Environment #2

Navigational Command	Lang2LTL Result	Code as Policies Result
1. navigate to the office with the wooden door, the classroom with glass door and the table in the atrium, kitchen counter, and the blue couch in any order	success	success
2. go down the hallway decorated with paintings, then find the kitchen table, then front desk, then staircase	success	success
3. navigate to classroom but do not visit classroom before the white table in atrium	success	misunderstand the task
4. only visit classroom once, and do not visit classroom until you visit elevator first	success	success
5. Go to the staircase, front desk and the white table in the atrium in that exact order. You are not permitted to revisit any of these locations	success	inexecutable
6. go to the purple elevator at least five times	success	inexecutable
7. visit the kitchen table at most three times	success	success
8. navigate to the classroom exactly four times	success	inexecutable
9. go to the front desk then the yellow office door, in addition do not visit the classroom with glass door	success	success
10. go to the stairs then the front desk, in addition avoid purple elevator	success	success
11. move to elevator then front desk, in addition avoid staircase	success	success
12. go to front desk exactly two times, in addition avoid elevator	success	inexecutable
13. Go to elevator, alternatively go to staircase	success	misunderstand the task
14. Go to the front desk at least two different occasions, in addition you are only permitted to visit the staircase at most once	success	misunderstand the task
15. Visit the elevator exactly once, in addition visit the front desk on at least 2 separate occasions	success	inexecutable
16. Go to the office, in addition avoid visiting the elevator and the classroom	success	success
17. Visit the front desk, in addition you are not permitted to visit elevator and staircase	success	success
18. Visit the purple door elevator, then go to the front desk and then go to the kitchen table, in addition you can never go to the elevator once you've seen the front desk	success	inexecutable
19. Visit the front desk then the white table, in addition if you visit the staircase you must avoid the elevator after that	success	inexecutable
20. Go to the classroom with glass door, but never visit the classroom with glass door	unsatisfiable. abort correctly	stop execution correctly
21. do not go to the white table until classroom, and do not go to the classroom until white table	unsatisfiable. abort correctly	stop execution correctly
22. go to kitchen table exactly once, in addition go to kitchen table at least twice	unsatisfiable. abort correctly	misunderstand the task
23. find the office, in addition avoid visiting the front desk and the classroom and the table in atrium	unsatisfiable. abort correctly	stop execution correctly
24. move to the kitchen table exactly twice, in addition pass by hallway decorated by paintings at most once	unsatisfiable. abort correctly	misunderstand the task
25. navigate to the kitchen table then the front desk, in addition after going to the kitchen table, you must avoid hallway decorated with paintings	unsatisfiable. abort correctly	misunderstand the task
26. Go to the front desk at least 4 different occasions, additionally, you are only permitted to visit the staircase at most once	incorrect grounding. OOD	inexecutable
27. Visit the front desk, additionally if you visit the elevator you must visit the office after that	incorrect grounding. OOD	success
28. Visit the front desk, additionally you visit the elevator you must visit the office after that the white table and the classroom	incorrect grounding. OOD	misunderstand the task

Table 4: Specification Patterns for Lang2LTL

Specification Type	Explanation	Formula
Visit	Visit a set of waypoints $\{p_1, p_2 \dots, p_n\}$ in any order	$\bigwedge_{i=1}^n \mathbf{F} p_i$
Sequence Visit	Visit a set of waypoints $\{p_1, p_2 \dots, p_n\}$, but ensure that p_2 is visited at least once after visiting p_1 , and so on	$\mathbf{F}(p_1 \wedge \mathbf{F}(p_2 \wedge \dots \wedge \mathbf{F}(p_n)) \dots)$
Ordered Visit	Visit a set of waypoints $\{p_1, p_2 \dots, p_n\}$, but ensure that p_2 is never visited before visiting p_1	$\mathbf{F}(p_n) \wedge \bigwedge_{i=1}^{n-1} (\neg p_{i+1} \mathbf{U} p_i)$
Strictly Ordered Visit	Visit a set of waypoints $\{p_1, p_2 \dots, p_n\}$, but ensure that p_2 is never visited before visiting p_1 , additionally, ensure that p_1 is only visited on a single distinct visit before completing the rest of the task	$\mathbf{F}(p_n) \wedge \bigwedge_{i=1}^{n-1} (\neg p_{i+1} \mathbf{U} p_i) \wedge \bigwedge_{i=1}^{n-1} (\neg p_i \mathbf{U} (p_i \mathbf{U} (\neg p_i \mathbf{U} p_{i+1})))$
Patrolling	Visit a set of waypoints $\{p_1, p_2 \dots, p_n\}$ infinitely often	$\bigwedge_{i=1}^n \mathbf{GF} p_i$
Bound Delay	If and only if the proposition a is ever observed, then the proposition b must hold at the very next time step	$\mathbf{G}(a \leftrightarrow \mathbf{X}b)$
Delayed Reaction	If the proposition a is ever observed, then its response is to ensure that the proposition b holds at some point in the future	$\mathbf{G}(a \rightarrow \mathbf{F}b)$
Prompt Reaction	If the proposition a is ever observed, then the proposition b must hold at the very next time step	$\mathbf{G}(a \rightarrow \mathbf{X}b)$
Wait	The proposition a must hold till the proposition b becomes true, and b may never hold	$a \mathbf{W} b$
Past Avoidance	The proposition a must not become true until the proposition b holds first. b may never hold	$\neg a \mathbf{W} b$
Future Avoidance	Once the proposition a is observed to be true, the proposition b must never be allowed to become true from that point onwards.	$\mathbf{G}(a \rightarrow \mathbf{XG}\neg b)$
Global Avoidance	The set of propositions $\{p_1, p_2 \dots, p_n\}$ must never be allowed to become true	$\bigwedge_{i=1}^n \mathbf{G}(\neg p_i)$
Upper Restricted Avoidance	The waypoint a can be visited on at most n separate visits	For $n = 1$, $\neg \mathbf{F}(a \wedge (a \mathbf{U} (\neg a \wedge (\neg a \mathbf{U} \mathbf{F}a))))$ For $n = 2$, $\neg \mathbf{F}(a \wedge (a \mathbf{U} (a \wedge (\neg a \mathbf{U} \mathbf{F}(a \wedge (a \mathbf{U} (\neg a \wedge (\neg a \mathbf{U} \mathbf{F}a))))))))$
Lower Restricted Avoidance	The waypoint a must be visited on at least n separate visits	For $n = 1$, $\neg \mathbf{F}a$ for $n = 2$, $\mathbf{F}(a \wedge (a \mathbf{U} (\neg a \wedge (\neg a \mathbf{U} \mathbf{F}a))))$
Exact Restricted Avoidance	The waypoint a must be visited on exactly n separate visits	For $n = 1$, $a \mathbf{M} (\neg a \vee \mathbf{G}(a \vee \mathbf{G}\neg a))$ For $n = 2$, $(a \wedge \mathbf{F}(\neg a \wedge \mathbf{F}a)) \mathbf{M} (\neg a \vee \mathbf{G}(a \vee \mathbf{G}(\neg a \vee \mathbf{G}(a \vee \mathbf{G}\neg a))))$