Applied Human Factors and Ergonomics (AHFE2025), Vol. 199, 2025, 230–238
https://doi.org/10.54941/ahfe1006835

Logic-Based Inference for Automated Compliance Checking on Indoor Accessibility Requirements

Jingyi Lai[1], Jiansong Zhang[1], Bradley Duerstock[2], and Nan Kong[2]

[1]School of Construction Management Technology, Purdue University, West Lafayette, IN, 47907, USA
[2]Weldon School of Biomedical Engineering and Edwardson School of Industrial Engineering, Purdue University, West Lafayette, IN, 47907, USA

ABSTRACT

Compliance checking for building designs based on Americans with Disabilities Act (ADA) requirements is mandatory to support accessibility. However, it is now mainly conducted manually and thus time-consuming and labour-intensive. Research has developed methods to support automated compliance checking (ACC) for building codes in separate phases. However, few of the existing work explored accessibility requirements. Moreover, they focused on individual phases without integrating them together to achieve ACC. In this research, we proposed a framework to achieve ACC for one requirement in ADA regarding accessible 180-degree turns. The framework first finds a path between any two elements in Building Information Modelling (BIM); after that it identifies if there are any 180-degree turns; and lastly the compliance checking outcome for the turn will be determined. In our experiment with 96 cases, the algorithm successfully identifies 180-degree turns and outputs compliance checking outcomes.

Keywords: Accessibility, Automated compliance checking, 180-degree turns, Building designs

INTRODUCTION

People with disabilities (PWD) need well-designed accommodation to have safe and comfortable access to various spaces. To ensure PWD's accessibility, Americans with Disabilities Act (ADA) was enforced when designing public accommodation and commercial facilities. However, manual compliance checking for ADA requirements is labour-intensive and time-consuming. To improve efficiency and consistency, automating the compliance checking process has become increasingly important.

Extensive research has been done on automated compliance checking (ACC) between building information modelling (BIM) and code requirements. Most of them share a similar methodology by: (1) retrieving and structuring required elements from BIM, (2) interpreting code requirements from natural language into programming language, (3) making the two types of information comparable, and (4) comparing the information to produce the compliance checking result. Some checking elements in the accessibility requirements are implicit, which means they cannot be directly

© 2025. Published by AHFE Open Access. All rights reserved.

retrieved from BIM data. For example, a route is not an explicit object, but comprises a space formed by nearby objects with certain boundaries. Also, a 180-degree turn cannot be retrieved directly from a BIM because it is defined by the geometric features of several routes. Since manual compliance checking of such requirements is effort demanding, this research aims to check the compliance of accessibility requirements on BIM automatically. To achieve this goal, it is essential to automate the analysis of such implicit elements on accessibility requirements.

The representation of implicit concepts and implementation of ACC are domain specific. It requires manual efforts to analyse and summarize suitable heuristics and develop corresponding ACC rules and processes for each domain. Even though some efforts have been made to automate the identification of implicit concepts such as egress, exit, and shaft (Wu & Zhang, 2022), little efforts have been made to support ACC in accessibility requirements regarding routes and turns.

Requirements for routes and turns are important parts in the accessibility requirements. To facilitate the ACC of accessibility requirements, this research starts from the identification of routes between objects in BIM. Once routes are identified, their parameters can also be extracted. Then, requirements related to routes and their parameters can be checked automatically. Particularly, this research focuses on the requirements regarding routes when there is a 180-degree turn.

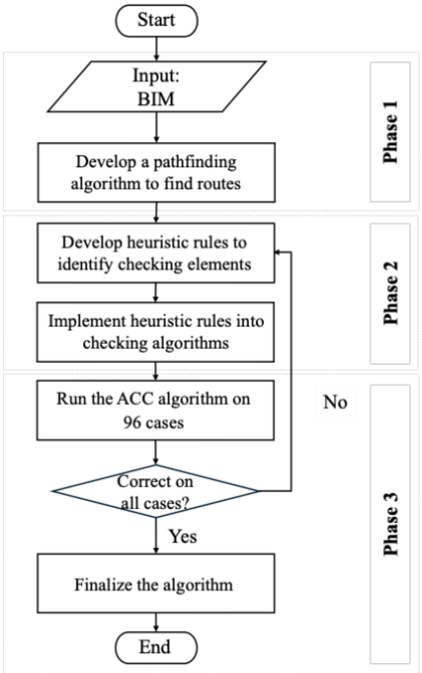

Figure 1: Framework to achieve ACC.

To achieve this, we broke down the process of pursuing the objective into three steps:

(1) finding a potentially accessible route between two objects in BIM;
(2) extracting to-be-checked concepts and attributes from BIM; and
(3) conducting compliance checking for each 180-degree turn in the route based on requirements.

We first developed a path finding algorithm for accessibility checking from BIM design based on A* algorithm. After that we extracted all the checking concepts and their associated attributes from BIM by developing heuristic rules and corresponding algorithm implementation. Lastly we developed an algorithm used the extracted information to conduct compliance checking for various routes. In an experiment with 96 testing cases, the algorithms were able to provide accurate assessment. The framework with three phases are shown in Figure 1.

METHOD

In this research, algorithms are developed to achieve our goals by processing an Industry Foundation Classes (IFC) file as input. First, a pathfinding algorithm is used to find a route between elements in the IFC file. Second, heuristic rules for each checking concept are developed and implemented into the corresponding algorithms, one algorithm for each rule. Third we combine all the developed algorithms to conduct ACC for each route regarding (1) whether there is a 180-degree turn; and (2) if so, the compliance checking result for the turn.

Finding Routes in BIM

We use an A*-based algorithm to find paths between objects in BIM. A* algorithm is commonly used to find the shortest path between a given start point and a goal point (Sathvik & Patil, 2021). The cost function of A* is shown in Eq. (1):

$$f(n) = g(n) + h(n) \tag{1}$$

where f(n)=total cost, g(n)=cost from the start to node n, h(n)=heuristic estimate of the cost from node n to the goal.

A* algorithm only considers distance as its costs when determining each step along the path. However, we also want to consider safety in our pathfinding algorithm. Therefore, the cost function of A* algorithm is tailored to find a path in BIM. More specifically, we add a penalty term that is inversely related to the clearance between node n and its nearest obstacle to the cost function g(n). In this way, g(n) is defined by both the cost from the start point to the current node n, and the clearance of n from its nearest obstacle. The h(n) stays the same as heuristic estimate of the cost from current node n to the goal.

Extracting Checking Concepts and Attributes From BIM

After finding a route between two objects in BIM, we aim to analyse this route according to ADA requirement. Heuristic rules are first developed to map the concepts in a requirement and the related objects in BIM. The heuristic rules are developed based on geometric, spatial, attribute-related information of entities as well as the relationships among entities in BIM. For instance, Wu & Zhang (2022) developed heuristic rules to extract egress from BIM. In their work, egress was defined as a door entity, with its location at the boundary of a building, and its relationship with other concepts as connecting interior and exterior. Then an algorithm is developed to incorporate the rules into programming languages by reviewing the human recognition process, summarizing inferable rules, and leveraging existing BIM data.

Compliance Checking for Each 180-Degree Turn

From the previous step we have identified all the checking concepts and their associated attributes to assess a 180-degree turn. The following step is to conduct compliance checking based on the information we have. To achieve that, we first classify compliance checking outcomes based on different scenarios. If there is not any 180-degree turn in the route, the algorithm will consider it as “no turn”. If there is a 180-degree turn, but its attributes fall into the exception which defines the types of turns that do not apply to the requirement, then it will be considered as “non-applicable”. If the turn is present and also applicable, but it does not comply to all the checking criteria, then it will be considered as “non-compliant”. If the turn is present and applicable and comply to all the checking criteria, then it will be considered as “compliant”.

Validation

To make sure the algorithm gives accurate result among various scenarios, we developed a dataset containing 96 cases to cover different scenarios. Three researchers were assigned to manually provide compliance checking results for the 96 cases, through: (1) classifying the compliance checking result for each case independently; and (2) comparing results and discuss them (i.e., convince each other) to achieve a consensus on the results of all cases. The manual results from the three researchers formed the gold standard of the compliance checking results among the 96 cases.

In this research, we use precision, recall and F1-measure to measure the performance of the algorithm. Precision is the ratio of true positives predicted by the algorithm to all the positives predicted by the algorithm [Eq. (2)] (Ren & Zhang, 2021; Wu & Zhang, 2022; Yang & Zhang, 2024; Zhang & El-Gohary, 2016). Recall measures the percentage of predicted true positives divided by all the actual positives [Eq. (3)] (Ren & Zhang, 2021; Wu & Zhang, 2022; Yang & Zhang, 2024; Zhang & El-Gohary, 2016).

$$P = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \tag{2}$$

$$R = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \tag{3}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{4}$$

EXPERIMENT AND RESULTS

We selected Section 403.5.2 in ADA (Department of Justice, 2010) to demonstrate the feasibility of the developed framework:

"*403.5.2 Clear Width at Turn. Where the accessible route makes a 180 degree turn around an element which is less than 48 inches (1220 mm) wide, clear width shall be 42 inches (1065 mm) minimum approaching the turn, 48 inches (1220 mm) minimum at the turn and 42 inches (1065 mm) minimum leaving the turn. EXC EPTION: Where the clear width at the turn is 60 inches (1525 mm) minimum compliance with 403.5.2 shall not be required.*"

The checking concepts and their attributes include:

(1) a route making a 180-degree turn;
(2) the width of the around-the-turn element;
(3) the clear width of the approaching-the-turn segment;
(4) the clear width of the at-the-turn segment; and
(5) the clear width of the leaving-the-turn segment.

We developed an algorithm to: (1) identify routes from selected elements, (2) extract checking concepts and attributes from BIM, and (3) conduct compliance checking.

Finding Routes in BIM

The algorithm took IFC files as input. It first projected each IFC model into a 2-dimensional (2D) view and annotated the centre of each boundary for elements as candidate start points and end points. Figure 2 shows an example of an IFC file and Figure 3 shows the route generated by the algorithm from point 13 to point 17.

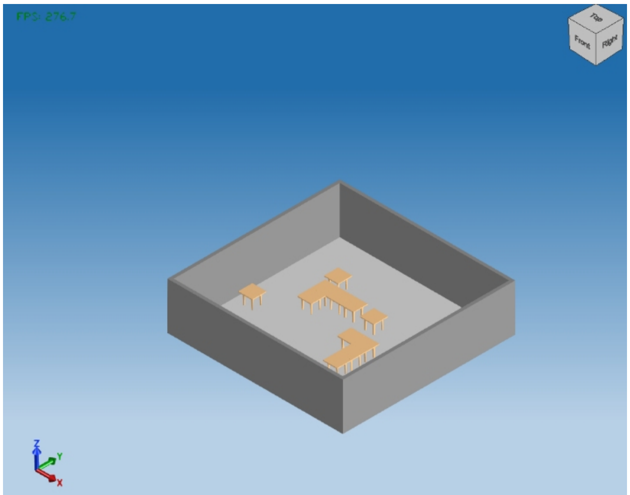

Figure 2: Example of an IFC file.

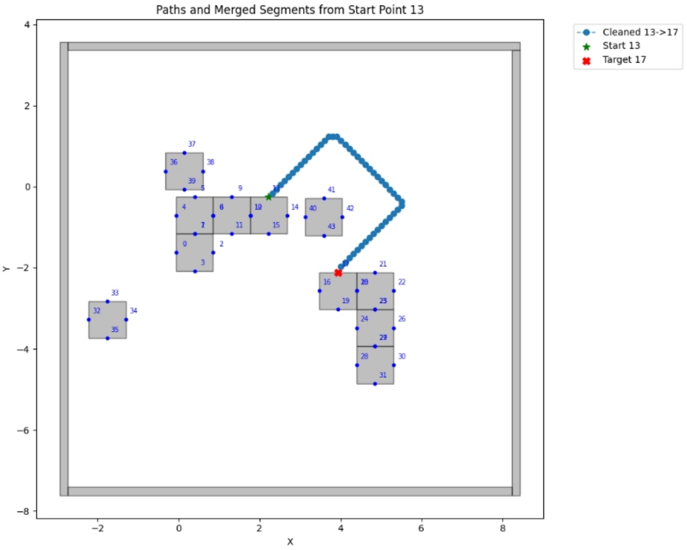

Figure 3: Route from point 13 to point 17.

Extracting To-Be-Checked Concepts and Attributes From BIM

Then we developed an heuristic-based algorithm to identify all the checking concepts and their attributes from BIM. As shown in Figure 4, the algorithm generated a report for all the checking concepts and their attributes. In the beginning, it recorded the start point and target point input by the user. Then it determined the route from point 13 to point 17 and calculated its total steps and length of the route. Then it identified that there was a 180-degree turn in the route. Next, it returned information regarding each segment in the route. At last, it produced a table showing the attribute values of checking concepts. It showed the width for the path around the element was 4'-1 12/16", the clear width for approaching-the-turn segment was 15'-8 15/16", the clear width of at-the-turn segment was 16'-9 1/16", the clear width of leaving-the-turn segment was 4'-4 6/16". In case there were missing segments, we considered all the other segments that had not been reported as "between segments" and calculated the minimum value for this type of segment. As shown in Figure 3, in this example, there was no such segment, and therefore the result was N/A.

Compliance Checking for Each 180-Degree Turn

Then we added the logic for determining compliance checking results for each detected 180-degree turn. The algorithm first determines if there is any 180-degree turn in a route. Then it determines the compliance checking result for each turn based on Section 403.5.2 of ADA. If the clear width of at-the-turn segment is more than 60 inches or the width of around-the-turn segment is equal to or more than 48 inches wide, then the turn will be determined as "non-applicable". If the turn is determined as applicable, then the algorithm continued to check if all the following criteria are met: (1) the clear width of approaching-the-turn segment is more than 42 inches wide; (2) the clear width of at-the-turn segment is more than 48 inches wide; (3) the clear width of leaving-the-turn segment is more than 42 inches wide. If all these four criteria are satisfied, then the turn will be considered a compliant 180-degree turn. If any one of the four criteria is not satisfied, then the turn will be considered as "non-compliant".

Figure 4 also shows the compliance checking result for the detected route in the column "status". The next column called "detail" shows it is not applicable because the clear width of at-the-turn segment is wider than 60 inches and the width of around-the-turn element is more than 48 inches.

```
Enter the index of the START furnishing point:
13

Enter the INDICES of TARGET furnishing points (comma-separated):
17

---Computing path 13 to 17---
Found path to 17, 51 steps; total length = 6.968 m  ≈ 22' - 10 6/16"

check all the path for whether there is a likewise 180 degree:
-Path to furnishing point 17 has a 180 degree turn.

return the result for compliance checking for 180 degree turn
segments: [((np.float64(2.3116766445719277), np.float64(-0.1609636980779552)), (np.float64(3.711676644571928), np.float64(1.2390363019220452))), ((np.float64(3.711676644571928), np.float64(1.2390363019220452)), (np.floa
t64(3.911676644571928), np.float64(1.2390363019220452))), ((np.float64(3.911676644571928), np.float64(1.2390363019220452)), (np.float64(5.511676644571929), np.float64(-0.36096369807795625))), ((np.float64(5.511676644571
929), np.float64(-0.36096369807795625)), (np.float64(5.511676644571929), np.float64(-0.4609636980779559))), ((np.float64(5.511676644571929), np.float64(-0.4609636980779559)), (np.float64(4.011676644571928), np.float64(-
1.960963698077956)))]
perp_run_seg = [(2, 2)]
Path to furnishing point 17 (WITH 180 degree turn) classified as: non-applicable

 Detailed Compliance Report for 180° Turns:
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
Path        Pair            AtTurn          Around(ft)        ApproachCW(ft)    AtCW(ft)          LeaveCW(ft)       MinBetween(ft)    Status            Detail
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
13->17      ((0, 1), (3, 4)) 2->2           4' - 1 12/16"     15' - 8 15/16"    16' - 9 1/16"     4' - 4 6/16"      N/A               non-applicable    AtCW 201.090 inches >= 60.0 inches AND around_width 49.77
0 inches >= 48.0 inches
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

Figure 4: Information extracted by the algorithm.

Validation

We used a dataset to help refine the algorithm. First, a dataset with 96 cases was developed, which were diversified in their compliance status with checking criteria specified in Section 403.5.2 of ADA and their geometric features. Table 1 shows all the features used to diversify the dataset.

Table 1: Features used to diversify the dataset.

Category	Attributes	Variation
Criteria checked in ADA	Clear width of at-the-turn segment	(1) < 48 inches (2) >= 48 inches and < 60 inches (3) > =60 inches
	Clear width of approaching-the-turn segment	(1) < 42 inches (2) >= 42 inches
	Clear width of leaving-the-turn segment	(1) < 42 inches (2) >= 42 inches
	Width of around-the-turn element	(1) < 48 inches (2) >= 48 inches

Continued

Table 1: Continued

Category	Attributes	Variation
Features mentioned in existing literature	Shape of the element in the start point	(1) horizontal quadrate (2) vertical quadrate (3) foursquare (4) L shape with the 90-degree angle toward southwest (5) L shape with the 90-degree angle toward southeast (6) L shape with the 90-degree angle toward northeast (7) L shape with the 90-degree angle toward northwest
	Shape of the element in the end point	(1) up (2) left (3) right (4) down (5) northwest (6) northeast (7) southeast (8) southwest
	Shape of the element around the turn	
	Orientation of the turn (direction the approaching-the-turn segment goes to)	
	Element coverage ratio in the room	(1) <=15% (2) >15% and <=50% (3) >50%
	Number of turns in the route	(1) 0 (2) 1 (3) >=2
	Length of the approaching-the-turn segment	(1) <=1.61 meters (2) <1.61 meters and >=2.15 meters (3) >2.15 meters

We ran the algorithm in each of the 96 cases iteratively. Every time the result was different from the gold standard, we identified the error and refined the algorithm until the result matched the gold standard. We repeated this process until the algorithm returned accurate results on all the 96 cases.

CONCLUSION

In this research, a framework was developed to help implement ACC for accessibility requirements, which: (1) finds routes between elements in BIM,

(2) identifies checking concepts together with their associated attributes, and (3) produces compliance checking results for each route. To show the feasibility of this framework, we implemented the framework in Section 403.5.2 of ADA in terms of 180-degree turns. We first adapted A* algorithm, tailoring it to find routes for wheelchair users in indoor environment. Then we used a heuristic-based algorithm to identify checking concepts and their attributes. At last, based on the checking logic in ADA, the algorithm could output if there was any 180-degree turn in the route and the compliance checking result for each 180-degree turn. We refined the algorithm in 96 cases iteratively. The algorithm gives accurate results in 96 cases in terms of their compliance checking results.

ACKNOWLEDGMENT

The authors would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 2348213. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

Department of Justice. (2010). *2010 ADA Standards for Accessible Design*. https://www.ada.gov/law-and-regs/design-standards/2010-stds/#403-walking-surfaces.

Ren, R., & Zhang, J. (2021). Semantic Rule-Based Construction Procedural Information Extraction to Guide Jobsite Sensing and Monitoring. *Journal of Computing in Civil Engineering*, *35*(6). https://doi.org/10.1061/(asce)cp.1943–5487.0000971.

Sathvik, N. G., & Patil, S. (2021). *Performance Analysis of Dijkstra's and the A-Star Algorithm on an Obstacle Map*. https://doi.org/10.1007/978–981-16–0171-2_8.

Wu, J., & Zhang, J. (2022). Model Validation Using Invariant Signatures and Logic-Based Inference for Automated Building Code Compliance Checking [Article]. *Journal of Computing in Civil Engineering*, *36*(3). https://doi.org/10.1061/(ASCE)CP.1943–5487.0001002.

Yang, F., & Zhang, J. (2024). Prompt-based automation of building code information transformation for compliance checking. *Automation in Construction*, *168*, 105817. https://doi.org/10.1016/J. AUTCON.2024.105817.

Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering*, *30*(2), 04015014. https://doi.org/10.1061/(ASCE) CP.1943–5487.0000346/ASSET/A1C0140B-7CC3–4CFD-B909–81A839AF321C/ASSETS/IMAGES/LARGE/FIGURE4. JPG.