SAT-based Fashion Store Problem Solver

Edoardo Ababei, Mykhailo Bozhko, Davide Frova, Mehmet Fatih Tekin
Group D

2025

1 Introduction

The Fashion Store problem involves systematically selecting garments and their associated colors based on precise constraints. The project objective is to leverage Boolean satisfiability (SAT) solving techniques to efficiently generate valid outfit combinations that meet predefined aesthetic and practical guidelines. This approach demonstrates the applicability and effectiveness of SAT in combinational decision-making scenarios.

2 Encoding into SAT

2.1 Variables

Each garment-color combination is encoded as a Boolean variable $x_{g,c}$, indicating whether a particular garment-color pair is selected or not. These variables are assigned unique integer identifiers to facilitate compatibility with SAT solver conventions.

2.2 Constraints

The constraints enforced by the SAT solver are categorized and detailed as follows:

- **Outfit Size Constraints**: The number of selected garments must be between 3 and 6 inclusive.
- **Type Coverage**: Each outfit must include at least one garment from every available category (hat, coat, top, bottom, shoes, gloves).
- **Palette Size Constraints**: The outfits must incorporate between 2 and 4 distinct colors.
- **Color Clashes**: Specific color combinations, notably red and pink, are disallowed.
- **Complement Harmony**: The selection of warm colors (red, orange, yellow) necessitates the inclusion of at least one cool color (blue, green, cyan).
- **Layering Order**: Certain garments require logical layering. For example, selecting a coat mandates the presence of a top garment beneath.
- **One-per-Body-Part**: Selection is limited to one garment per body part category.
- **Seasonal Constraints (Winter)**: The presence of outerwear, specifically a coat or gloves, is mandatory for the winter context.
- **Style Preferences (Soft Constraints)**: Preference-based constraints are incorporated, such as prioritizing black gloves over white gloves, blue coats over red coats, and black tops over green tops.

2.3 CNF Conversion

All constraints are systematically translated into Conjunctive Normal Form (CNF). For example, a color clash constraint between a red hat and pink top is encoded as follows:

$$(\neg x_{hat,red} \vee \neg x_{top,pink})$$

3 Implementation

3.1 Software and Libraries

The implementation utilizes the Z3 solver due to its robustness and efficiency. Python scripts manage the translation of constraints into CNF format, input parsing, solver invocation, and result interpretation. The web interface employs Flask for backend functionality and Bootstrap for frontend styling and responsiveness.

3.2 Solver Integration

The pipeline for solver integration includes:

1. Parsing user input provided via direct text entry or file uploads.
2. Generating CNF clauses based on parsed inputs and encoded constraints.
3. Executing the SAT solver (Z3) with generated clauses.
4. Parsing solver outputs to produce user-friendly results.

4 User Interface

The interactive web interface is designed using Flask and provides:

- Facilities for file uploads and direct textual input of garment specifications.
- Quick test functionalities allowing users to test SAT and UNSAT scenarios.
- Dynamic exploration and visualization of constraints.
- Clear and readable solver result presentations.

5 Experiments and Results

5.1 Testing Methodology

Rigorous testing procedures included various predefined SAT and UNSAT scenarios designed to thoroughly validate the solver's correctness and reliability.

5.2 Results

Experiments demonstrated exceptional solver performance, showing rapid and accurate garment selection within milliseconds. The solver reliably identified unsatisfiable combinations, providing immediate feedback.

6 Challenges and Alternatives

Encountered challenges primarily involved:

- The complexity of effectively encoding intricate constraints into CNF without performance degradation.
- Ensuring responsiveness and usability of the graphical user interface.

Alternatives such as different encoding strategies and solver technologies were evaluated, but the selected Z3 solver consistently outperformed these alternatives.

7 Conclusion

The project successfully demonstrated the power of SAT-based approaches for solving complex combinational problems in the context of fashion selection. The solution provided not only efficiency and robustness in constraint satisfaction but also offered an intuitive and user-friendly interface. Future developments may include enhanced management of dynamic soft constraints and broader customization capabilities to cater to evolving user preferences.