- according to the matching relationship of each sc-element between each sc-construction pair with the same number in the semantic graph of the standard answer and the semantic graph of the user answer, the mapping relationships of potential equivalent variable sc-nodes pairs between the semantic graphs are established.

Fig. 2 shows an example of establishing the mapping relationship between semantic graphs in SCg-code.

In Fig. 2, the definition of the partial ordering relation is described. A binary relation R is called a partial ordering, or partial order if and only if it is: reflexive, antisymmetric and transitive.

When the mapping relationship between the potential equivalent variable sc-node pairs between the semantic graphs is established, the similarity between answers can be calculated, and the detailed calculation process is shown below:

- decomposing the semantic graphs of the answers into substructures according to the structure of the knowledge description;
- establishing the mapping relationship of potential equivalent variable sc-node pairs between the semantic graphs;
- using formulas (1), (2) and (3) to calculate the precision, recall and similarity between semantic graphs.

If the similarity between semantic graphs is not equal 1, it is also necessary to determine whether their logical formulas are logically equivalent. Because any predicate logic formula has a PNF equivalent to it. Therefore, based on the approach to convert predicate logic formulas into PNF and characteristics of logic formulas in ostissystems, an approach to convert logic formulas into unique (deterministic) PNF according to strict restriction rules is proposed in this article [23], [24]. The strict restrictions mainly include the following:

- renaming rule is preferred when converting logical formulas to PNF;
- existential quantifier is moved to the front of the logical formula in preference;
- the logical formula can usually be expressed in the following form: $(Q_1x_1Q_2x_2...Q_nx_n(\mathrm{A} \iff \mathrm{B}))$, where $Q_i(i = 1, ...n)$ is a quantifier. A is used to describe the definition of a concept at a holistic level, and it does not contain any quantifiers. B is used to explain the semantic connotation of a definition at the detail level, and it is usually a logical formula containing quantifiers [8], [24]. Therefore, in order to simplify the knowledge processing, it is only necessary to convert the logical formula B to PNF;

The process of converting the semantic graph constructed based on logic formula into PNF descriptions is shown below:

- if there are multiple sc-structures connected by the same conjunctive connective, the sc-constructions contained in them are merged into the same scstructure;
- eliminating all the implication connectives;
- moving all negative connectives to the front of the corresponding sc-structure;
- using renaming rules so that all bound variables in the semantic graphs are not the same;
- moving all quantifiers to the front of the logical formula;
- merging again the sc-structures in the semantic graphs that can be merged.

If the calculated similarity between the semantic graphs of PNF representation is not 1, the similarity between the semantic graphs calculated for the first time is used as the final answer similarity.

Fig. 3 shows an example of converting a semantic graph into PNF representation in SCg-code.

In Fig. 3, the definition of the reflexive relation is described. In mathematics, a binary relation R on a set M is reflexive if it relates every element of M to itself

Calculating the similarity between answers to proof questions and problem-solving task

Both proof questions and problem-solving task follow a common task-solving process:

1) the set (Ω) of conditions consisting of some known conditions;
2) deriving an intermediate conclusion using some of the known conditions in Ω and adding it to Ω. Each element in Ω can be regarded as a solving step;
3) repeat step 2) until the final result is obtained [25], [26].

This task-solving process is abstracted as a directed graph, whose structure is in most cases an inverted tree, and is called a reasoning tree (i. e. the reasoning tree of the standard answer). The automatic verification process of user answers to this type of test questions is the same as the traditional manual answer verification process, i.e., verifying whether the current solving step of the user answer is a valid conclusion of the partial solving step preceding that step. This means whether the solving step in the user answer corresponding to the parent node in the reasoning tree always is located after the solving steps in the user answer corresponding to the child nodes [27].

The semantic graphs of user answers to proof questions and problem-solving task in the ostis-systems are linear structures consisting of some semantic sub-graphs for describing the solving steps and some semantic fragments for describing the logical order and transformation processes between the semantic sub-graphs. The semantic graph of standard answers to this type of test questions is an reasoning tree consisting of a number of search templates (which can be abstracted as the nodes in the tree). Each search template is constructed using SCL-