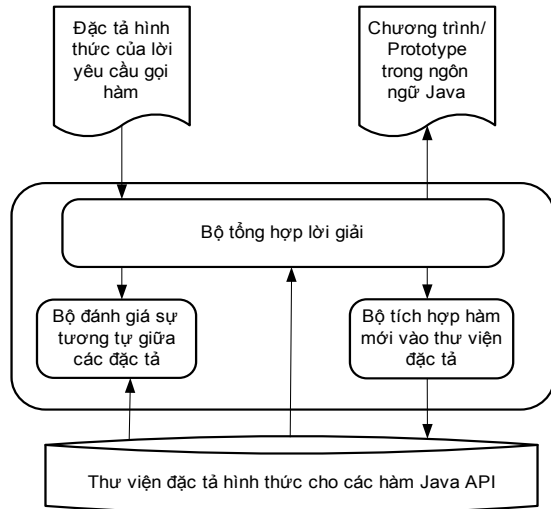


Hàm đặc trưng $\varphi: \mathcal{C} \rightarrow \{0,1\}^m$, với mỗi c thuộc \mathcal{C} thì chúng ta có vector $\varphi^c \in \{0,1\}^m$ thỏa điều kiện $\varphi_i^c = 1 \Leftrightarrow \Phi(c,i) = 1$.



Hình 7: Khung thức cho việc tìm kiếm và tái sử dụng hàm API tự động

Định nghĩa 4: Hàm phân loại

Cho mỗi p trong \mathcal{P} , hàm phân loại được định nghĩa như sau $C_p(\cdot): \mathcal{C} \rightarrow \mathcal{R}$, với mỗi lời yêu cầu c thì hàm phân loại $C_p(c)$ sẽ tính toán mức độ phù hợp của hàm p so với lời yêu cầu c .

Ví dụ, với các hàm f_1, f_2 và f_3 được mô tả trong Hình 8, nếu chúng ta ký hiệu các mệnh đề p_1, p_2, p_3, p_4 tương ứng với các tập toán tử $\{+, -\}$, $\{*\}$, $\{>, \leq\}$, $\{\log\}$ ¹, chúng ta sẽ có ma trận đặc trưng như sau:

φ	p_1	p_2	p_3	p_4
f_1	0	1	0	0
f_2	1	0	0	0
f_3	1	0	1	1

```
//@ ensures \result == 2 * x;
double f1 (double x);

//@ ensures \result == - x;
double f2 (double x);

//@ ensures
//@ (x>0)==>(\result==x+1) &&
//@ (x<0)==>(\result==log(x2));
double f3 (double x);
```

Hình 8: Một số đặc tả hàm bằng JML

¹ Đây là một sự phân loại toán tử thường gặp dựa trên độ ưu tiên,

Khi đó, chúng ta sẽ tính được vector đặc trưng cho từng lời gọi hàm dựa vào φ . Ví dụ dựa vào ma trận trên, chúng ta có vector đại diện cho lời gọi hàm f_2 là $\varphi_{f_2} = [1, 0, 0, 0]$ (giá trị 1 ở cột thứ i nghĩa là c_2 có đặc trưng i). Với vector φ vừa tính được, chúng ta có thể phát triển các phương pháp tính toán độ tương tự. Các phương pháp này có thể đơn giản là phép tích góc giữa hai vector, hoặc có thể là một phương pháp học máy như Naive Bayes [27], tùy theo độ phức tạp của bài toán.

3.2 Bộ tổng hợp lời giải

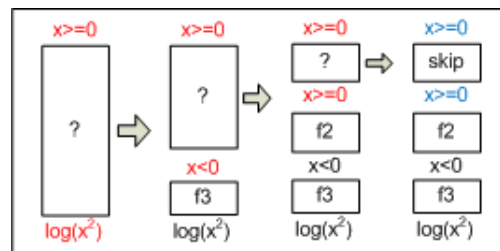
Khởi chức năng thứ hai là bộ tổng hợp lời giải. Đây là bộ phận quan trọng nhất trong khung thức. Nhờ vào việc các hàm thư viện được đặc tả bằng các hàm logic và phương pháp tính độ tương tự giữa các đặc tả xây dựng được, chúng ta có thể phát triển một cơ chế tìm kiếm và tổng hợp các hàm API đáp ứng nhu cầu ban đầu dựa trên kỹ thuật *lập kế hoạch* trong lĩnh vực Trí tuệ Nhân tạo (AI Planning).

Bộ tìm kiếm hàm sử dụng kỹ thuật lập kế hoạch dựa trên trí tuệ nhân tạo hoạt động theo nguyên tắc sau: Xuất phát từ yêu cầu ban đầu (c) và tập các hàm P có độ tương tự trong ngưỡng quy định đối với (c), bộ lập kế hoạch sẽ chọn ra một hàm p từ P (theo độ ưu tiên về mức độ phù hợp), nếu tiên điều kiện của p phù hợp với tiên điều kiện của c thì quá trình tìm kiếm kết thúc và trả kết quả về. Ngược lại, giải thuật xem tiên điều kiện của p như là một hậu điều kiện mới và lặp lại quá trình tìm kiếm. Trong trường hợp quá trình tìm kiếm bế tắc, giải thuật sẽ quay lui lại với các hàm p' trong P ($p' \neq p$).

Với các hàm thư viện như mô tả trong Hình 8, giả sử chúng ta có một lời yêu cầu gọi hàm được đặc tả như sau:

```
//@ requires x >= 0
//@ ensures \result == log(x2)
```

Với thư viện hàm và lời yêu cầu như trên, quá trình *lập kế hoạch* của chúng ta được thực hiện thông qua các bước như trong Hình 9.



Hình 9: Quá trình tìm kiếm hàm dựa trên lập kế hoạch