

0. Riferimenti.

- Hassan Ait-Kaci
"Warren's Abstract Machine - A tutorial reconstruction"
MIT Press, 1999
(wambook.pdf ?)

1. Prolog --> WAM-code.

2. Organizzazione logica della WAM.

- Code;
- Heap;
- Stack : Environment + Choice-point/Backtrack control;
- Trail;
- PDL.

3. Registri principali.

- Code : P, CP;
- Heap : H;
- Stack : E, B;
- Trail : TR;
- PDL : impliciti;
- Entry : Ai (argomenti), Xj (variabili).

4. Operazioni principali.

- Heap : deref, bind, unify;
- Trail : trail, unwind_trail;
- PDL : reset, empty, push, pop;
- Cntrl : fail.

5. Istruzioni.

- Operazioni sui termini:
set_var, set_val, put_str, put_var, put_str,
uni_var, uni_val, get_str, get_var, get_str;
- Operazioni di controllo:
call, proceed, alloc, dealloc,
try_me_else, retry_me_else, trust_me.

6. Rappresentazione dei termini.

- Variabili libere;
- Variabili istanziate;
- Riferimenti (entry point) a strutture;
- Strutture: funtori, arita' e regole di contiguita';
- Esempi: s(s(z)), ... seq(S, while(not(C),S)).

7. Operazioni base relative alla rappresentazione dei termini.

- deref;
- bind;
- unify.

8. Codice per la costruzione dei termini nello Heap.

- Struttura del codice e compilazione: costruzione 'bottom-up';
- Simulazione dell'esecuzione del codice per la costruzione della rappresentazione dei termini nello Heap;
- Pseudo-codice di put_str, set_var, set_val.

9. Rappresentazione dei termini e unificazione.

- Esempio: seq(assign(x,E), while(G,I)).
- Soluzione preliminare: costruzione della rappresentazione dei due termini e invocazione dell'unificatore;
- Esercizio: simulazione dell'esecuzione del codice preliminare;
- Esempio limite: unificazione di due termini 'quasi' ground:
while(not(G), assign(x,e)).
?- while(not(g), assign(x,E)).
- Struttura di codice specifico per l'unificazione e compilazione: analisi 'top-down' e ruolo del registro argomento A1;
- Simulazione dell'esecuzione del codice per l'unificazione.

10. Raffinamento del processo di unificazione tramite istruzioni.

- Modalita' 'Write' e 'Read' a confronto;
- 'Write': costruzione top-down di un template con slot=variabili + eventuali operazioni di istanziazione delle variabili;
- 'Read': confronto top-down dei termini riferiti dagli entry-point registrati;
- Pseudo-codice di get_str, uni_var, uni_val;
- Simulazione dell'esecuzione del codice relativo all'esempio iniziale.

11. Interpretazione procedurale dei predicati.

- Passaggio di termini come argomenti;
- Pseudo-codice di put_var, put_val, get_var, get_val;
- Struttura del codice e compilazione;
- Esempio:
eq(repeat(S,C), seq(S,while(not(C),S))).
?- eq(repeat(assign(x,E),G), P).
- Pseudo-codice delle istruzioni call, proceed (provvisorio).

12. Controllo 'procedurale'.

- Esempio:
div(X, s(Y), Q, R) :- mul(Q, s(Y), M), add(M, R, X), les(R, s(Y)).
- Organizzazione dell'environment frame;
- Struttura del codice e compilazione;
- Pseudo-codice delle istruzioni alloc, dealloc.