Constraint Handling Rules

cf. tutorial by Tom Schrijvers on first day of ICLP'08
Turing-completeness
CHR is Turing-complete...

... but which (syntactic) subclasses of CHR are still Turing-complete and which are not?

in other words: which features of CHR are really needed and which are redundant?

(from the computability point of view)
Host language

- CHR extends host language (e.g. Prolog); constraint arguments are of host language data type
- **Propositional CHR**: no host language, no arguments (only arity 0 constraints)
- **CHR-only**: no host language, arguments are constants or variables (that cannot be bound)
- **Sufficiently strong** host language: complex terms are available, we can do arithmetic
propositional CHR
(no constraint arguments)

CHR-only
(no host language)

CHR with
sufficiently strong
host language

CHR with
Turing-complete
host language
CHR-only (no host language)

CHR with sufficiently strong host language

CHR with Turing-complete host language

propositional CHR (no constraint arguments)

not Turing-complete

Turing-complete
propositional CHR
(no constraint arguments)

CHR-only
(no host language)

CHR with
sufficiently strong
host language

CHR with
Turing-complete
host language

CHR’05 result:
RAM machine simulator in CHR
TMSIM:

r1 @ delta(Q,S,P,T,left), adj(L,C) \ state(Q), cell(C,S), head(C)
   => state(P), cell(C,T), head(L).

r2 @ delta(Q,S,P,T,right), adj(C,R) \ state(Q), cell(C,S), head(C)
   => state(P), cell(C,T), head(R).

r3 @ delta(Q,S,P,T,left) \ left(C), state(Q), cell(C,S), head(C)
   => cell(L,b), left(L), adj(L,C), state(P), cell(C,T), head(L).

r4 @ delta(Q,S,P,T,right) \ right(C), state(Q), cell(C,S), head(C)
   => cell(R,b), adj(C,R), right(R), state(P), cell(C,T), head(R).

Also in CHR'05 paper:
Turing machine simulator in CHR-only
Kinds of rules

- Different kinds of rules:
  - **propagation** rules: $\text{KeptHead} \Rightarrow \text{Body}$
  - **simplification** rules: $\text{RemovedHead} \Leftrightarrow \text{Body}$
  - simpagation rules: $\text{KeptHead} \setminus \text{RemovedHead} \Leftrightarrow \text{Body}$

- Simplification rules can simulate all rules
  “only simplification rules” = “only simpagation rules” = “any kind of rule”

- “Programs with only propagation rules” vs “Programs with any kind of rule”
propositional CHR
(no constraint arguments)

CHR-only
(no host language)

CHR with
sufficiently strong
host language

CHR with
Turing-complete
host language

prop. -only

simp. -only

TMSIM
propositional CHR
(no constraint arguments)

CHR-only
(no host language)

CHR with
sufficiently strong host language

CHR with
Turing-complete host language

prop. only

simp. only

TMSIM

TMSIM

-PROP
**TMSIM-PROP:**

% add timestamps

head(C) ==> inittime(T), head(T,C).

inittime(T), state(Q) ==> state(T,Q).
inittime(T), cell(C,S) ==> cell(T,C,S).
inittime(T), adj(L,R) ==> adj(T,L,R).

% compute next step

r13 @ state(T,Q), head(T,C), cell(T,C,S), delta(Q,S,Q2,S2,left) ==> next(T,U), state(U,Q2), cell(U,C,S2), mleft(T,C,U), cright(T).
r24 @ state(T,Q), head(T,C), cell(T,C,S), delta(Q,S,Q2,S2,right) ==> next(T,U), state(U,Q2), cell(U,C,S2), mright(T,C,U), cleft(T).

state(T,Q), head(T,C), cell(T,C,S), nodelta(Q,S), reject(Q) ==> fail.

% move head, extending tape if needed

mleft(T,C,U), adj(T,L,C) ==> L \== null \ mid head(U,L), cleft(T).
mleft(T,C,U), adj(T,null,C) ==> head(U,L), adj(U,null,L), adj(U,L,C).
mright(T,C,U), adj(T,C,R) ==> R \== null \ mid head(U,R), cright(T).

% copy non-modified tape to next timestamp

cell(T,C,S), next(T,U), head(T,C2) ==> C \== C2 \ mid cell(U,C,S).
adj(T,L,R), next(T,U) ==> L \== null, R \== null \ mid adj(U,L,R).
cleft(T), next(T,U), adj(T,X,null) ==> adj(U,X,null).
cright(T), next(T,U), adj(T,null,X) ==> adj(U,null,X).
Multi-headed vs. single-headed

- Head of a rule is a *conjunction* of constraints
  - 1 conjunct: **single-headed** rule
  - 2 conjuncts: two-headed rule
  - > 1 conjuncts: **multi-headed** rule
- >2-headed rule can be written using 2-headed rules:
  - “a, b, c, d ⇒ e” can be written as follows:
    - “a, b ⇒ ab”    “ab, c ⇒ abc”    “abc, d ⇒ e”
propositional CHR
(no constraint arguments)

CHR-only
(no host language)

CHR with
sufficiently strong
host language

CHR with
Turing-complete
host language

single-headed
prop. only
simp. only

multi-headed
prop. only
simp. only

TMSIM

TMSIM - PROP

TMSIM

TMSIM

TMSIM

TMSIM
MINSKY-A:
see Di Giusto, Gabbrielli, Meo (2008): “Expressiveness of multiple heads in CHR”
MINSKY-A:
see Di Giusto, Gabbrielli, Meo (2008): “Expressiveness of multiple heads in CHR”
Number of rules

- Do we really need more than one rule?
<table>
<thead>
<tr>
<th>Type of Rule</th>
<th>Single-Headed</th>
<th>Multi-Headed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only One Rule</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Propositional CHR (no constraint arguments)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR-only (no host language)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR with sufficiently strong host language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR with Turing-complete host language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple Rules</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single-Headed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple-Headed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMSIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minsky</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR with sufficiently strong host language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Propositional CHR (no constraint arguments)</td>
<td>CHR-only (no host language)</td>
<td>CHR with sufficiently strong host language</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>only one rule</td>
<td></td>
<td></td>
</tr>
<tr>
<td>single-headed</td>
<td>single-headed</td>
<td>single-headed</td>
</tr>
<tr>
<td>prop. only</td>
<td>prop. only</td>
<td>prop. only</td>
</tr>
<tr>
<td>simp. only</td>
<td>simp. only</td>
<td>simp. only</td>
</tr>
<tr>
<td>multiple rules</td>
<td></td>
<td></td>
</tr>
<tr>
<td>single-headed</td>
<td>single-headed</td>
<td>single-headed</td>
</tr>
<tr>
<td>prop. only</td>
<td>prop. only</td>
<td>prop. only</td>
</tr>
<tr>
<td>simp. only</td>
<td>simp. only</td>
<td>simp. only</td>
</tr>
<tr>
<td>TMSIM</td>
<td>TMSIM -1R</td>
<td>TMSIM</td>
</tr>
<tr>
<td>MINSKY</td>
<td>MINSKY -A</td>
<td>MINSKY</td>
</tr>
<tr>
<td>TMSIM</td>
<td>TMSIM</td>
<td>TMSIM</td>
</tr>
</tbody>
</table>
CHR-only (no host language)  
CHR with sufficiently strong host language  
CHR with Turing-complete host language  

propositional CHR (no constraint arguments)

only one rule

multiple rules

single-headed

multi-headed

prop. only

simp. only

TMSIM - 1R

TMSIM - PROP

TMSIM

MINSKY - A

single-headed

multi-headed

prop. only

simp. only

prop. only

simp. only

prop. only

simp. only

prop. only

simp. only
TMSIM-2R:
% left(C) <=> adj(L,L), adj(L,C), cell(L,b).
% right(C) <=> adj(R,R), adj(C,R), cell(R,b).

r13 @ delta(Q,S,P,T,left), state(Q), head(C) \ adj(L,C), adj(C,R), cell(C,S)
    <=> adj(L,C2), adj(C2,R), adj(C,C), cell(C,b), cell(C2,T), state(P), head(L).

r24 @ delta(Q,S,P,T,right), state(Q), head(C) \ adj(L,C), adj(C,R), cell(C,S)
    <=> adj(L,C2), adj(C2,R), adj(C,C), cell(C,b), cell(C2,T), state(P), head(R).

TMSIM-1R:
% adj(A,B) <=> adj(A,B,left), adj(B,A,right).

r1234 @ delta(Q,S,P,T,D), state(Q), head(C) \ adj(A,C,D), adj(C,B,D),
                  cell(C,S), adj(C,A,E), adj(B,C,E)
    <=> adj(A,C2,D), adj(C2,B,D), adj(C,C,D), cell(C,b), cell(C2,T),
               adj(C2,A,E), adj(B,C2,E), adj(C,C,E), state(P), head(A).
Operational semantics

- **abstract semantics**: rules are applied in any order (so program has to be confluent)

- **refined semantics**: specific execution order (cf. Prolog: left-to-right, depth-first + triggering / “active constraint”)

**propositional CHR**
- Abstract semantics
- Refined semantics

**CHR-only**
- (no host language)

**CHR with sufficiently strong host language**

**CHR with Turing-complete host language**

### only one rule
- single-headed
  - propositional only
  - simple only
- multi-headed
  - propositional only
  - simple only

### multiple rules
- single-headed
  - propositional only
  - simple only
- multi-headed
  - propositional only
  - simple only

- Single-headed multi-headed
  - TMSIM
  - -1R
  - -PROP
  - TMSIM

- Multi-headed
  - MINSKY
  - -A
propositional CHR

(no constraint arguments)

abstract semantics

refined semantics

CHR-only
(no host language)

CHR with sufficiently strong host language

CHR with Turing-complete host language

only one rule

single-headed

prop. -only

prop. -only

prop. -only

prop. -only

TMSIM -1R

TMSIM -PROP

TMSIM

multiple rules

single-headed

prop. -only

prop. -only

prop. -only

prop. -only

MINSKY -A

MINSKY -PROP

MINSKY -P
<table>
<thead>
<tr>
<th></th>
<th>only one rule</th>
<th>multiple rules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>propositional CHR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(no constraint</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>arguments)</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>abstract semantics</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>refined semantics</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>CHR-only</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>(no host language)</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>CHR with</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>sufficiently strong</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>host language</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>CHR with</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>Turing-complete</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
<tr>
<td>host language</td>
<td><img src="image.png" alt="Table" /></td>
<td><img src="image.png" alt="Table" /></td>
</tr>
</tbody>
</table>
Conclusion

- one multi-headed rule is enough
- single-headed rules are not enough \(\text{(unless we have a sufficiently strong host language)}\)
- propositional CHR is not enough, \textit{unless} we can use the refined semantics (!!)
- more information:
  - my PhD thesis (chapter 10.3)