Online appendix for the paper

Parallel Instantiation of ASP Programs: Techniques and Experiments

published in Theory and Practice of Logic Programming

Simona Perri, Francesco Ricca, Marco Sirianni
Dipartimento di Matematica, Università della Calabria, 87030 Rende, Italy
(*e-mail:* {perri, ricca, sirianni}@mat.unical.it)

submitted 28 November 2010; revised 16 April 2011; accepted 31 August 2011

Appendix A Splitting the Extension of a Literal

In Figure A1 is detailed an implementation of procedure *SplitExtension*, which plays a central role in the single-rule parallelization algorithms presented in Section 3.2. In particular, function *SplitExtension* partitions the extension of l (stored in S and ΔS) into n splits. In order to avoid useless copies, each split is virtually identified by means of iterators over S and ΔS, representing ranges of instances.

```
Procedure SplitExtension ($l$: Literal; $n$:integer; $S$: SetOfAtoms; $\Delta S$: SetOfAtoms,
          var vector<VirtualSplit> $V$)
begin
    integer $size$:= $\lfloor$ ($S$.size() + $\Delta S$.size())/ $n$ $\rfloor$;
    integer $i$:= 0;   AtomsIterator $it$ := $S$.begin();
    while $i < \lfloor S$.size()/$size\rfloor$ do   // possibly, build splits with atoms from S
        $V[i]$.SetIterators_S($it$, $it$+$size$);   $it$ := $it$ + $size$;   $i$=$i$+1;
    end while
    if $it < S$.end() then   // possibly, build a split mixing S and $\Delta S$ atoms
        $V[i]$.SetIterators_S($it$, $S$.end());   $it$ := $\Delta S$.begin();
        integer $k$ := $size$ - Size($V[i]$);
        if $\Delta S$.size() < $k$
            $V[i]$.SetIterators_$\Delta S$ ($it$, $\Delta S$.end());   $it$ = $\Delta S$.end();
        else
            $V[i]$.SetIterators_$\Delta S$ ($it$, $it$+$k$);   $it$ = $it$+$k$;   $i$ = $i$+1;
    while $i < \lfloor(S$.size()+$\Delta S$.size())/$size\rfloor$ do   // possibly, build splits with atoms from $\Delta S$
        $V[i]$.SetIterators_$\Delta S$ ($it$, $it$+$size$);   $it$ := $it$ + $size$;   $i$=$i$+1;
    end while
    if $it < \Delta S$.end() then
        $V[i]$.SetIterators_$\Delta S$ ($it$, $\Delta S$.end());
end Procedure
```

Fig. A 1. Splitting the extension of a literal.

More in detail, for each split, an instance of *VirtualSplit* is created containing two iterators over S (resp. ΔS), namely S_begin and S_end (resp. ΔS_begin,ΔS_end), indicating the instances of l from S (resp. ΔS) that belong to the split. The procedure starts by building splits with atoms from S; then, it proceeds by considering atoms from ΔS. Note that, in general, a split may mix ground atoms from both S and ΔS.

Appendix B Detailed Results: 3Colorability and Hamiltonian Path

Tables B 1 and B 2 contain detailed results for the benchmarks 3Colorability and Hamiltonian Path. We recall that, in the case of 3Colorability, were generated 18 simplex graphs by means of the Stanford GraphBase library using the function $simplex(n, n, -2, 0, 0, 0, 0)$, where $80 \leq n \leq 250$; whereas, for the Hamiltonian Path benchmark, 14 graphs were generated by using a tool by Patrik Simons, with n nodes with $1000 \leq n \leq 12000$.

In detail, Table B 1 reports the results of an experimental analysis aimed at comparing the effects of the single rule parallelism with the first two levels. The first column reports the problem considered, whereas the next columns report the results for four version of the instantiator: (i) `serial` where parallel techniques are not applied, (ii) $\texttt{levels}_{1+2}$ where components and rules parallelism are applied, (iii) $\texttt{level}_3$ where only the single rule level is applied, and (iv) $\texttt{levels}_{1+2+3}$ in which all the three levels are applied.

Table B 2 reports the results of a scalability analysis on the instantiator $\texttt{levels}_{1+2+3}$, which exploits all the three parallelism levels. In particular, both the average instantiation times and the efficiencies are reported by considering the effects of increasing both the size of the instances and the number of available processors (from 1 up to 8 CPUs).

Problem	`serial`	`levels`$_{1+2}$	`level`$_3$	`levels`$_{1+2+3}$
$3-Col_1$	8.61 (0.00)	8.55 (0.09)	1.27 (0.03)	1.26 (0.03)
$3-Col_2$	14.16 (0.00)	13.70 (0.04)	1.90 (0.01)	1.90 (0.01)
$3-Col_3$	20.44 (0.00)	20.08 (0.10)	2.85 (0.12)	2.79 (0.03)
$3-Col_4$	30.92 (0.00)	29.58 (0.17)	4.09 (0.01)	4.28 (0.30)
$3-Col_5$	42.13 (0.00)	41.36 (0.06)	5.62 (0.01)	5.67 (0.13)
$3-Col_6$	59.38 (0.00)	57.08 (0.24)	8.03 (0.21)	7.81 (0.03)
$3-Col_7$	78.64 (0.00)	79.19 (0.07)	10.29 (0.11)	10.36 (0.15)
$3-Col_8$	104.45 (0.00)	102.99 (0.34)	13.70 (0.04)	13.65 (0.04)
$3-Col_9$	133.72 (0.00)	132.11 (1.96)	17.58 (0.11)	17.47 (0.08)
$3-Col_{10}$	177.28 (0.00)	171.71 (0.84)	23.00 (0.08)	22.70 (0.37)
$3-Col_{11}$	220.07 (0.00)	218.34 (0.75)	29.40 (0.23)	28.85 (0.05)
$3-Col_{12}$	281.29 (0.00)	281.61 (0.65)	37.22 (0.30)	37.33 (0.28)
$3-Col_{13}$	347.97 (0.00)	346.69 (1.65)	45.48 (1.35)	45.88 (1.12)
$3-Col_{14}$	420.88 (0.00)	432.82 (2.02)	59.86 (2.46)	58.39 (0.74)
$3-Col_{15}$	528.20 (0.00)	536.62 (1.00)	72.39 (3.07)	71.35 (0.44)
$3-Col_{16}$	673.24 (0.00)	644.07 (2.99)	85.75 (1.22)	86.46 (0.63)
$3-Col_{17}$	786.43 (0.00)	784.43 (1.00)	103.16 (0.95)	102.44 (3.54)
$3-Col_{18}$	965.36 (0.00)	966.00 (2.31)	120.09 (2.18)	124.70 (1.08)
HP_1	0.07 (0.00)	0.05 (0.00)	0.03 (0.00)	0.03 (0.00)
HP_2	3.50 (0.00)	3.21 (0.00)	0.56 (0.01)	0.57 (0.01)
HP_3	13.24 (0.00)	12.42 (0.01)	1.96 (0.01)	1.97 (0.04)
HP_4	29.28 (0.00)	27.85 (0.06)	4.21 (0.10)	4.18 (0.05)
HP_5	51.80 (0.00)	49.28 (0.16)	7.38 (0.13)	7.33 (0.03)
HP_6	80.87 (0.00)	77.24 (0.05)	11.56 (0.07)	11.56 (0.09)
HP_7	117.16 (0.00)	112.35 (0.56)	16.29 (0.02)	16.77 (0.23)
HP_8	160.79 (0.00)	154.58 (0.45)	22.44 (0.15)	22.45 (0.16)
HP_{10}	212.84 (0.00)	205.60 (0.39)	29.23 (0.07)	29.57 (0.10)
HP_{11}	274.43 (0.00)	263.61 (1.11)	37.60 (0.24)	37.14 (0.29)
HP_{12}	343.05 (0.00)	331.16 (0.50)	46.63 (0.09)	47.68 (0.03)
HP_{13}	422.72 (0.00)	406.36 (0.64)	57.54 (0.48)	57.66 (1.05)
HP_{14}	510.15 (0.00)	492.28 (1.68)	68.95 (0.56)	70.26 (0.38)

Table B 1. *3Colorability and Hamiltonian Path results: Average instantiation times in seconds (standard deviation)*

Problem	serial	2 proc	3 proc	4 proc	5 proc	6 proc	7 proc	8 proc	2 proc	3 proc	4 proc	5 proc	6 proc	7 proc	8 proc
$3-Col_1$	8.61 (0.00)	4.41 (0.02)	3.02 (0.02)	2.29 (0.03)	1.86 (0.03)	1.59 (0.02)	1.38 (0.03)	1.26 (0.03)	0.98	0.95	0.94	0.93	0.90	0.89	0.85
$3-Col_2$	14.16 (0.00)	7.15 (0.02)	4.83 (0.01)	3.65 (0.03)	2.93 (0.00)	2.51 (0.04)	2.19 (0.03)	1.90 (0.01)	0.99	0.98	0.97	0.97	0.94	0.92	0.93
$3-Col_3$	20.44 (0.00)	10.49 (0.01)	7.11 (0.06)	5.35 (0.02)	4.31 (0.03)	3.68 (0.11)	3.12 (0.04)	2.79 (0.03)	0.97	0.96	0.96	0.95	0.93	0.94	0.92
$3-Col_4$	30.92 (0.00)	15.74 (0.27)	10.78 (0.19)	7.97 (0.02)	6.35 (0.02)	5.37 (0.03)	4.77 (0.20)	4.28 (0.30)	0.98	0.96	0.97	0.97	0.96	0.93	0.90
$3-Col_5$	42.13 (0.00)	21.76 (0.43)	14.58 (0.20)	10.93 (0.21)	8.86 (0.12)	7.46 (0.15)	6.57 (0.17)	5.67 (0.13)	0.97	0.96	0.96	0.95	0.94	0.92	0.93
$3-Col_6$	59.38 (0.00)	29.96 (0.05)	19.94 (0.00)	15.08 (0.07)	12.16 (0.11)	10.15 (0.07)	8.87 (0.14)	7.81 (0.03)	0.99	0.99	0.98	0.98	0.98	0.96	0.95
$3-Col_7$	78.64 (0.00)	40.25 (0.19)	26.66 (0.42)	20.31 (0.19)	16.24 (0.34)	13.44 (0.13)	11.56 (0.10)	10.36 (0.15)	0.98	0.98	0.97	0.97	0.98	0.97	0.95
$3-Col_8$	104.45 (0.00)	53.79 (0.33)	35.70 (0.35)	26.85 (0.36)	21.45 (0.28)	17.96 (0.18)	15.55 (0.14)	13.65 (0.04)	0.97	0.98	0.97	0.97	0.97	0.96	0.96
$3-Col_9$	133.72 (0.00)	68.53 (0.22)	46.32 (0.17)	34.79 (0.34)	27.60 (0.18)	23.34 (0.13)	20.18 (0.19)	17.47 (0.08)	0.98	0.96	0.96	0.97	0.95	0.95	0.96
$3-Col_{10}$	177.28 (0.00)	89.29 (0.65)	60.65 (0.58)	44.91 (0.24)	36.34 (0.26)	30.54 (0.51)	26.35 (0.06)	22.70 (0.37)	0.99	0.97	0.99	0.98	0.97	0.96	0.98
$3-Col_{11}$	220.07 (0.00)	113.03 (1.76)	76.25 (0.54)	57.71 (0.92)	45.77 (0.52)	38.94 (0.44)	32.97 (0.37)	28.85 (0.05)	0.97	0.96	0.95	0.96	0.94	0.95	0.95
$3-Col_{12}$	281.29 (0.00)	143.00 (2.39)	98.02 (0.38)	73.68 (0.84)	59.08 (0.27)	49.56 (0.72)	42.73 (0.17)	37.33 (0.28)	0.98	0.96	0.95	0.95	0.95	0.94	0.94
$3-Col_{13}$	347.97 (0.00)	178.98 (4.07)	123.22 (2.62)	90.09 (1.79)	71.84 (1.48)	61.42 (2.13)	53.32 (0.52)	45.88 (1.12)	0.97	0.94	0.97	0.97	0.94	0.93	0.95
$3-Col_{14}$	420.88 (0.00)	224.85 (3.82)	152.87 (2.07)	113.88 (0.92)	92.51 (1.66)	76.46 (0.57)	66.68 (0.40)	58.39 (0.74)	0.94	0.92	0.92	0.91	0.92	0.90	0.90
$3-Col_{15}$	528.20 (0.00)	278.26 (5.67)	184.32 (1.48)	138.84 (1.63)	111.29 (2.58)	94.25 (0.97)	80.58 (0.24)	71.35 (0.44)	0.95	0.96	0.95	0.95	0.93	0.94	0.93
$3-Col_{16}$	673.24 (0.00)	331.55 (4.12)	224.66 (4.57)	169.90 (7.72)	137.37 (1.34)	111.57 (2.10)	97.50 (0.56)	86.46 (0.63)	1.02	1.00	0.99	0.98	1.01	0.99	0.97
$3-Col_{17}$	786.43 (0.00)	403.02 (5.80)	263.37 (2.69)	207.01 (5.21)	158.80 (5.81)	135.89 (1.64)	116.68 (3.89)	102.44 (3.54)	0.98	1.00	0.95	0.99	0.96	0.96	0.96
$3-Col_{18}$	965.36 (0.00)	473.35 (4.88)	312.04 (6.90)	238.99 (4.15)	190.61 (1.65)	159.41 (4.64)	140.46 (1.49)	124.70 (1.08)	1.02	1.03	1.01	1.01	1.01	0.98	0.97
HP_1	3.50 (0.00)	1.82 (0.01)	1.27 (0.00)	1.00 (0.02)	0.80 (0.00)	0.70 (0.00)	0.63 (0.00)	0.57 (0.01)	0.96	0.92	0.88	0.88	0.83	0.79	0.77
HP_2	13.24 (0.00)	6.84 (0.05)	4.64 (0.01)	3.59 (0.05)	2.90 (0.04)	2.48 (0.01)	2.19 (0.06)	1.97 (0.04)	0.97	0.95	0.92	0.91	0.89	0.86	0.84
HP_3	29.28 (0.00)	15.63 (0.28)	10.41 (0.26)	7.90 (0.19)	6.30 (0.02)	5.52 (0.22)	4.67 (0.03)	4.18 (0.05)	0.94	0.94	0.93	0.93	0.88	0.90	0.88
HP_4	51.80 (0.00)	26.55 (0.03)	18.05 (0.02)	13.82 (0.16)	11.14 (0.04)	9.52 (0.14)	8.22 (0.05)	7.33 (0.03)	0.98	0.96	0.94	0.93	0.91	0.90	0.88
HP_5	80.87 (0.00)	42.60 (0.18)	28.68 (0.38)	21.61 (0.05)	17.56 (0.12)	15.07 (0.13)	13.05 (0.01)	11.56 (0.09)	0.95	0.94	0.94	0.92	0.89	0.89	0.87
HP_6	212.84 (0.00)	110.90 (0.48)	74.50 (0.56)	56.16 (0.26)	45.62 (0.35)	38.10 (0.42)	33.46 (0.11)	29.57 (0.10)	0.96	0.95	0.93	0.92	0.91	0.89	0.87
HP_7	274.43 (0.00)	141.00 (0.59)	94.69 (0.84)	72.07 (0.11)	58.30 (0.35)	48.86 (0.15)	42.67 (0.21)	37.14 (0.29)	0.97	0.96	0.94	0.94	0.92	0.90	0.90
HP_8	160.79 (0.00)	82.52 (0.35)	56.06 (0.23)	42.64 (0.34)	34.09 (0.11)	29.26 (0.05)	25.50 (0.32)	22.45 (0.16)	0.96	0.95	0.95	0.93	0.93	0.91	0.90
HP_9	343.05 (0.00)	176.40 (0.89)	118.84 (0.47)	89.88 (0.53)	73.53 (0.19)	61.61 (0.43)	53.33 (0.10)	47.68 (0.03)	0.97	0.97	0.95	0.94	0.94	0.92	0.92
HP_{10}	117.16 (0.00)	60.84 (0.29)	41.05 (0.49)	31.40 (0.32)	25.55 (0.13)	21.43 (0.17)	18.81 (0.08)	16.77 (0.23)	0.97	0.96	0.95	0.93	0.93	0.92	0.90
HP_{11}	422.72 (0.00)	218.38 (0.51)	146.26 (0.38)	110.34 (0.56)	89.77 (0.19)	75.15 (0.25)	66.01 (0.59)	57.66 (1.05)	0.97	0.96	0.96	0.94	0.94	0.91	0.92
HP_{12}	510.15 (0.00)	261.06 (2.47)	173.57 (1.93)	132.88 (0.21)	107.51 (0.63)	90.66 (0.80)	78.44 (0.44)	70.26 (0.38)	0.98	0.98	0.96	0.95	0.94	0.93	0.91

Table B 2. *3Colorability and Hamiltonian Path results: average instantiation times in seconds (standard deviation), efficiency*

Appendix C Application of the Single Rule Parallelism

In the following is reported a complete example of the application of the single rule parallelism for computing in parallel the instantiation of a rule. Consider the following program $\mathcal{P}$ encoding the 3-Colorability problem:

$$\begin{array}{ll} (r) & col(X, red) \lor col(X, yellow) \lor col(X, green) \text{ :- } node(X). \\ (c) & \text{:- } edge(X, Y), col(X, C),\ col(Y, C). \end{array}$$

Assume that, after the instantiation of rule r, the extensions of predicate $node$ and predicate col, are the ones reported in Table C 1, and that the extension of the predicate $edge$ is the one reported in Table C 2.

	Predicates extension
1	$node(a)$
2	$node(b)$
3	$node(c)$
4	$node(d)$
1	$col(a, red)$
2	$col(a, yellow)$
3	$col(a, green)$
4	$col(b, red)$
5	$col(b, yellow)$
6	$col(b, green)$
7	$col(c, red)$
8	$col(c, yellow)$
9	$col(c, green)$
10	$col(d, red)$
11	$col(d, yellow)$
12	$col(d, green)$

Table C 1. *Extension of the predicate* $node$ *and col.*

Suppose now that the heuristics suggests to perform the single rule level of parallelism for the instantiation of the constraint (c), and suppose that the extension of predicate $edge$ is split in two. Then, the extension of the predicate $edge$ is partitioned into two subsets which appear divided by an horizontal line in Table C 2. The instantiation of constraint (c) is carried out in parallel by two separate processes, say p_1, and p_2, which will consider as extension of $edge$, respectively, the two splits depicted in Table C 2. Process p_1 produces the following ground constraints:

$$\begin{array}{l} \text{:-}\, edge(a, b), col(a, red),\ col(b, red). \\ \text{:-}\, edge(a, b), col(a, yellow),\ col(b, yellow). \\ \text{:-}\, edge(a, b), col(a, green),\ col(b, green). \\ \text{:-}\, edge(b, c), col(a, red),\ col(b, red). \\ \text{:-}\, edge(b, c), col(a, yellow),\ col(b, yellow). \\ \text{:-}\, edge(b, c), col(a, green),\ col(b, green). \end{array}$$

	Predicate extension
1	$edge(a,b)$
2	$edge(b,c)$
3	$edge(b,d)$
4	$edge(c,d)$

Table C 2. *Extension of the predicate* $edge$*.*

whereas, process p_2 produces the following ground constraints:

$$
\begin{array}{l}
:\!-\, edge(b,d), col(b,red),\ col(d,red).\\
:\!-\, edge(b,d), col(b,yellow),\ col(d,yellow).\\
:\!-\, edge(b,d), col(b,green),\ col(d,green).\\
:\!-\, edge(c,d), col(c,red),\ col(d,red).\\
:\!-\, edge(c,d), col(c,yellow),\ col(d,yellow).\\
:\!-\, edge(c,d), col(c,green),\ col(d,green).
\end{array}
$$

Tecnically speaking, this is obtained by a call to procedure *SplitExtension* described in Appendix A. The procedure will create two *virtual splits*, say V_1 and V_2, with:

$$
\begin{array}{ll}
V_1.S_{begin} = edge(a,b) & V_1.\Delta S_{begin} = \perp\\
V_1.S_{end} = edge(b,c) & V_1.\Delta S_{end} = \perp\\
V_2.S_{begin} = edge(b,d) & V_2.\Delta S_{begin} = \perp\\
V_2.S_{end} = edge(c,d). & V_2.\Delta S_{end} = \perp
\end{array}
$$

where $\perp$ indicates a null iterator (usually indicating an iterator that has moved after the end of a container), which, in this case, it is used to represent that that no split is created containing instances from ΔS.