```
-- $Id: par-paraffins.sa,v 1.2 1995/06/12 08:14:52 borisv Exp borisv $

-- $Log: par-paraffins.sa,v $
-- Revision 1.2  1995/06/12 08:14:52  borisv
-- Added some comments
--
-- Revision 1.1  1995/06/12 06:05:25  borisv
-- Initial revision
--

-- a parallel version of the paraffins problem
-- need everything in the serial version
-- The only difference is the contructor below

class PAR_PARAFFINS < $PARAFFINS is
    include PARAFFINS;

    create(n:INT):SAME is
        -- Create all paraffin isomers of size up to n.
        -- Trivial parallelization: performs all iterations
        -- of molecule-constructing loops of paraffins in parallel.
        -- This is possible because the list of molecules of
        -- size i is completely independent of any other molecules
        -- of sizes defferent from i.
        -- Moreover, as soon as we finsh generating all radicals
        -- needed for molecules of a particular size, we start
        -- generating such molecules without waiting for completion
        -- of the entire radical list (this is possible because
        -- molecules of size i are built from radicals no bigger than
        -- (i-1)/2)

        tmp1, tmp2 : ARRAY{INT};

        r ::= new;
        r.radicals := #(n/2+1);
        r.molecules := #(n+1);

        -- initialize radicals

        r.radicals[0] := r.radicals[0].push(#HYDROGEN_RADICAL);
        -- hydrogen is a radical of size 0 (no C atoms)
        r.generate_molecules_of_size(1);

        -- parallel part starts here.
        parloop
        -- first, generate radicals
        -- This has to be mostly serial as bigger
        -- radicals use smaller ones. There is some
        -- limited parallelizm here, as we can try building
        -- bigger radicals when only a portion of the smaller
        -- ones is available. If we were to exploit it here,
        -- this would significantly complicate the logic while
        -- giving only a small performance improvement if any,
        -- as by far most of work is in generating molecules from
        -- radicals. Also, paralelization of the generation
        -- of molecule would become a real challenge. Interestingly
        -- enough, none of the solutions even mentioned this possibility.

             i ::= 1.upto!(n/2);
             r.gen_radicals_of_size(i);
        do
           -- lists of paraffin molecules of
           -- different sizes are generate in parallel
```

```
            -- generate paraffin molecules
            -- a separate thread is created for each list of
            -- of molecules with a unique size
             r.generate_molecules_of_size(2*i);
            if 2*i < n then
                r.generate_molecules_of_size(2*i+1);
            end;
       end; -- parloop
       return r;
    end;  -- create

    generate_molecules_of_size(size:INT) is
        -- generate paraffin isomer molecules
        -- of size "size"

        -- generate carbon centered molecules for both odd
        -- and even size paraffin molecules.
        gen_ccps_of_size(size);
        if (size % 2) = 0 then
            -- even size molecule: need bond centered paraffins as well
            gen_bcps_of_size(size);
        end;
    end;
end; -- class PAR_PARAFFINS
```